Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments

Lin Gu [1] Dong Jia [2] Pascal Vicaire [1] Ting Yan [1] Liqian Luo [1]
Ajay Tirumala [3] Qing Cao [1] Tian He [1]
John A. Stankovic [1] Tarek Abdelzaher [1] Bruce H. Krogh [2]

{lingu, pv9f, ty4k, ll4p, qc3h, th7c, stankovic, zaher}@cs.virginia.edu
{djia, krogh}@ece.cmu.edu tirulama@uiuc.edu

ABSTRACT

A wide variety of sensors have been incorporated into a spectrum of wireless sensor network (WSN) platforms, providing flexible sensing capability over a large number of low-power and inexpensive nodes. Traditional signal processing algorithms, however, often prove too complex for energy-and-cost-effective WSN nodes. This study explores how to design efficient sensing and classification algorithms that achieve reliable sensing performance on energy-and-cost-effective hardware without special powerful nodes in a continuously changing physical environment. We present the detection and classification system in a cutting-edge surveillance sensor network, which classifies vehicles, persons, and persons carrying ferrous objects, and tracks these targets with a maximum error in velocity of 15%. Considering the demanding requirements and strict resource constraints, we design a hierarchical classification architecture that naturally distributes sensing and computation tasks at different levels of the system. Such a distribution allows multiple sensors to collaborate on a sensor node, and the detection and classification results to be continuously refined at different levels of the WSN. This design enables reliable detection and classification without involving high-complexity computation, reduces network traffic, and emphasizes resilience and adaptation to the realistic environment. We evaluate the system with performance data collected from outdoor experiments and field assessments. Based on the experience acquired and lessons learned when developing this system, we abstract common issues and introduce several guidelines which can direct future development of detection and classification solutions based on WSNs.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.3 [**Computer System Organization**]: Special Purpose And Application-Based Systems—*Real-Time and embedded systems*; C.4 [**Performance of Systems**]: Design Studies

SenSys'05, November 2–4, 2005, San Diego, California, USA.
Copyright 2005 ACM 1-59593-054-X/05/0011 ...$5.00.

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Classification, Wireless Sensor Networks, VigilNet

1. INTRODUCTION

Sensing is a fundamental function in wireless sensor networks. Researchers have built WSN platforms with a wide spectrum of sensors, ranging from simple thermistors to micropower impulse radars [4, 13, 14]. They provide flexible sensing capability with a large number of low-power and inexpensive sensor nodes. Nontrivial as it is, the selection and integration of sensors on a WSN platform is often a manageable task given a certain amount of engineering effort. The situation is, however, completely different above the physical sensor and computing hardware layer – the acquisition and processing of sensor data impose great challenges on WSN design because of strict resource constraints.

Cost-effectiveness being an important objective, WSN designers often choose mass produced commercial off the shelf (COTS) sensors when designing a sensor network system. Moreover, a sensor node must be energy efficient. As a result, the raw sensor data is often of low-quality – they are not always reliable, not always repeatable, usually not self-calibrated, and often not shielded to environment and circuit board noise. Obviously, it is necessary to use signal processing algorithms to filter, process, and abstract sensor data with software to provide precise, reliable, and easy-to-use information to applications.

Traditional signal processing algorithms, however, often prove too complex to implement on inexpensive sensor network hardware without digital signal processing co-processors. For example, the popular Berkeley Mica series has an 8-bit micro-processor running at 7.3827MHz, no hardware floating-point support, and only 4KB data memory. Though recent versions of MicaZ and Telos motes

[1] Department of Computer Science, University of Virginia.

[2] Department of Electrical and Computer Engineering, Carnegie Mellon University.

[3] Department of Computer Science, University of Illinois at Urbana-Champaign

[4] This work is supported in part by NSF grant CCR-0098269 and CCR-0325197, the MURI award N00014-01-1-0576 from ONR, and the DAPRPA IXO offices under the NEST project (grant number F336615-01-C-1905).

employ a bigger data memory, we expect the growth of computational resources on WSN platforms to be rather slow because of the emphasis on low power consumption, low cost, and small form factor. Generally, resource constraints will continue to represent the reality of energy-and-cost-effective embedded systems. This strict resource limitation makes it very difficult to execute Fast Fourier Transformation (FFT) and other signal processing algorithms with moderate or high time/space complexity. Also, the stringent energy budget favors simple and quick algorithms over complex algorithms that require prolonged execution time.

While the computation/energy resources are limited, the application requirement is not. Specifically, the development of recent surveillance WSNs requires the network to provide functionalities well beyond sensing and routing. Such surveillance WSNs are designed to detect and report certain classes of events of interest. When such an event happens, the WSN needs to detect it quickly, classify it into one category (e.g., person, vehicle), and compute its attributes (e.g., location, velocity).

Designing such surveillance WSNs is a research challenge. Besides the obviously severe resource constraints, the following factors also contribute to the difficulty of the task.

- To provide sensing coverage for a relatively large area, the network is usually comprised of a large number of densely deployed nodes. This imposes a challenge on efficient data propagation and reliable operation.
- The detection, classification, and reporting must be performed in a timely manner. It is usually required that the network complete the detection and classification before the target travels out of the field so that the system can respond to the event. As a result, offline-style processing performed by base stations with global and relatively "complete" data is often not feasible in this context.
- To perform quality signal processing, the sensors often need to sample at a high sampling rate, stressing resource utilization. The sensing data is bursty and in large quantity.
- Surveillance networks are often deployed on rough terrains for a long period of time. Hence, it must be adaptive to the realistic, ever-changing environment.

Given the numerous technical challenges, important research questions are: Can we construct a reliable surveillance WSN that meets the requirements within the strict resource constraints? What performance will such a system achieve? This study attempts to answer these questions by presenting the detection and classification system in VigilNet [10], which is a recently deployed surveillance WSN detecting and classifying vehicles, persons and persons with ferrous objects. Specifically, this paper explores the design choices involved in constructing an efficient detection and classification system that achieves reliable performance on a network of energy-and-cost-effective sensor nodes, analyzes the performance, and proposes a set of guidelines for future designs of WSNs in a similar design context.

It is worth clarifying that advanced signal processing mathematics and algorithms are not the emphasis of this paper. Instead, this paper focuses on the system design issues involved in creating a reliable and realistic classification system for a surveillance WSN using homogeneously low-end sensor nodes, as well as evaluation of the effectiveness of these designs. To the authors' best knowledge, there has not been a large-scale deployment of such a sophisticated surveillance network without using special powerful nodes. Hence, our study is focused on answering this challenge: Without enhancing any individual nodes' capability and cost, can a network of distributed sensor nodes provide advanced functions and work reliably in realistic environments? We believe that the experience acquired and lessons learned in constructing such a system, and the analysis of the trade-offs and design decisions in it, will benefit the research in this area, and help transform the research potential of WSNs into real-world technology and market success.

The paper is organized as follows. Section 2 presents background information and surveys related work. Section 3 gives an overview of the VigilNet surveillance system. Section 4 presents the design of the hierarchical classification architecture. System level evaluation is shown in Section 5. Section 6 discusses several guidelines for designing a large-scale WSN for detection and classification tasks. Finally, Section 7 concludes the paper.

2. BACKGROUND

Focusing on VigilNet's hardware platform, we present a brief overview of the sensing subsystem – sensors and their supporting circuitry – on a sensor node. The sensing subsystem is the hardware foundation on which classification systems are constructed. We also survey the related work in the area of detection and classification WSN systems.

2.1 Overview of the sensing subsystem

VigilNet uses the ExScal motes as sensor nodes. Based on the Mica2 [3] mote design, the ExScal mote, shown in Fig. 1, is designed by CrossBow Inc. and Ohio State University for large-scale surveillance WSNs [8]. The major difference between the ExScal mote and the Berkeley Mica2 mote is that the former integrates a magnetometer (Honeywell HMC1052[2]), a microphone, and 4 PIR sensors on the same circuit board as the processor's. After the first prototype ExScal motes were delivered in March 2004, CrossBow released several versions with various improvements throughout the year of 2004.

Several correlated factors contribute to the complexity of the sensing circuitry. First, applications require a long sensing distance, which implies a finer granularity for the sensor readings. Second, as a general purpose platform, designers hope to choose sensors with a wide measuring range. Third, the wider measuring range combined with a finer granularity maps to more numeric values which, however, have to all fall into the representation capability of the A/D converter (ADC), I/O bus, and CPU I/O port width. Finally, as the sensors on the sensor board grow in both number and sophistication, the support circuitry may need to support better filtering, handle more advanced signaling protocols, employ a faster or wider bus, provide wider functionality (such as waking up the sensor node), or build better shielding to avoid cross-talk among various components. These factors make the design of the sensing subsystem a significant engineering effort involving numerous design choices which often depend on the application domain.

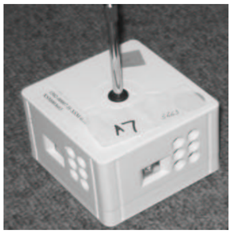


Figure 1: ExScal mote

To solve the aforementioned range and granularity problems for the magnetometer, the ExScal mote includes circuitry that allows the application program to adjust the input signal to be amplified. To provide a quality signal for acoustic processing, the microphone circuitry incorporates a high-pass filter and a low-pass filter. Both the input adjustment and filtering are controllable by the processor, with an I^2C bus connecting the processor and sensor components.

2.2 Related work

With the development of WSN systems, sensing, detection, and tracking have been a prosperous research area. Specifically, Wang et. al. studied acoustic tracking using Mica motes [22]. Simon et. al. designed a sniper localization system with acoustic signal processing [19] and accomplished good performance. Different from VigilNet's homogeneous approach, these systems employ special powerful nodes or DSP co-processors to process acoustic data. Zhao et. al. described collaborative signal processing [25] to retrieve more accurate information from sensor data and achieve better target tracking performance. Pattem et. al. build a framework to evaluate the tracking strategies in an energy aware context [18]. Most of the performance analysis in [25] and [18] are conducted by simulations, concentrating on exploring the design space and trade-offs under specific constraints and assumptions.

Along the direction of real-world application and deployments, researchers have also constructed a number of successful systems. Szewczyk et. al. [21] developed a habitat monitoring WSN on the Great Duck Island and the system operated for months. Zhang et. al. developed a WSN for wild life tracking [24]. These systems demonstrate the flexibility and capability of the WSN technology in various applications. However, VigilNet faces more demanding application requirements. As a result, many design choices are different in these systems than in VigilNet. For example, many current systems typically employ centralized processing which is not feasible in many surveillance networks [7].

In [11], the authors describe a surveillance network that can detect moving targets. The system uses Mica2 motes [3] equipped with a magnetometer (Honeywell HMC1002 [2]), an acoustic sensor and, on some nodes, a motion sensor. The motion sensor is an Advantaca MIR (micropower impulse radar) sensor which transmits microwave signals and detects motion by capturing distortion of the reflected signal. The network reports a target as a walking person or a vehicle. Therefore, it has a preliminary classification capability. However, there is very limited signal processing in it. As a result, the classification is limited in both functionality and performance. Also, the MIR sensors, worth four thousand dollars each, are not a typical choice for energy-and-cost-effective systems.

Brooks et. al. [7] introduced a collaborative signal processing framework for sensor networks using location-aware routing and collaborative signal processing. Their study provides many insights into the distributed collaborative classification in WSNs. Nevertheless, the CSP framework involves non-trivial training and computation overhead, which our system cannot afford. Also, the system implementation and evaluation of the CSP framework employ nodes with higher power than the energy-and-cost-effective WSN nodes our system is targeting. In fact, VigilNet must satisfy three conflicting requirements simultaneously – low-end hardware, long lifetime, and sophisticated function. This challenging design context is different than what past solutions assume.

Among recently deployed WSNs, the Extreme Scaling project is the most similar to VigilNet in functionality and hardware platform [1, 8]. However, a major difference is that the Extreme Scaling WSN employs a heterogeneous network topology and uses a more powerful Stargate node for some computation and communication intensive tasks.

3. OVERVIEW OF THE DETECTION AND CLASSIFICATION IN VIGILNET

The VigilNet surveillance system [5] is a WSN with 200 sensor nodes (ExScal motes). The WSN is required to perform timely detection, tracking and classification of vehicles, persons, and persons with ferrous objects. When a target is detected, the WSN reports the detection to an external device. The external device can be a more powerful sensor, a communication device connecting to a control center, or any device that handles the information delivered by the WSN. A base mote connects to the external device through a UART interface, and serves as a router between the WSN and the external device. As the target travels in the network, the WSN garners enough information to classify the target and compute its attributes, such as location and velocity, and the results are delivered to the external device as periodic updates. Fig. 2 shows a screen snapshot of VigilNet deployed along two roads forming a "T" shape. It illustrates the detection and classification of a "person with ferrous objects" target. Moreover, the WSN is to be deployed in a rough terrain and operate for months. Hence, the detection and classification algorithms must be adaptive to environmental variety and weather changes.

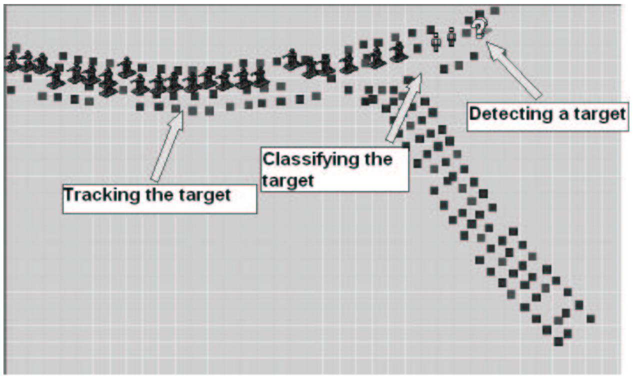


Figure 2: Screen of tracking a person with ferrous objects

As in many surveillance systems, VigilNet emphasizes that the false negative rate (the possibility of a target not being detected) must be very low. Meanwhile, it also requires a low false positive rate (the possibility of an event being reported without a real target in the field) since false positives waste energy and reduce the overall system lifetime. This implies that the wake-up (most of the network nodes are in sleep mode when there are no events of interest), sensing and classification must complete within a time constraint. These two factors – energy efficiency and low latency – make it undesirable to have a centralized semi-offline algorithm that collects all data from the network, transports them to a base station, and lets a powerful node analyze data and perform classification. Instead, the network, including the base mote (also an energy-and-cost-effective device), must perform reliable detection and classification functions independently in a timely manner without powerful nodes involved.

To build a complete VigilNet for realistic outdoor environments, other middleware services are also integrated. In brief, the localization is done through the walking GPS solution [20], which assigns nodes their location at the time they are deployed. The time synchronization used in VigilNet is a variation of the FTSP protocol [17] without periodic adjustments for the sake of stealthiness. Routing infrastructure is a set of multi-parent diffusion trees (forest) rooted at the base nodes. To achieve long-term surveillance, a multi-dimensional power management scheme is proposed in [12]. In this paper, we focus on the design of the detection and classification system in VigilNet, which is not addressed in other papers, but is a major part of the system and directly determines the system's functionality and performance.

4. CLASSIFICATION SYSTEM DESIGN

In this section, we present the design of the classification architecture, including the sensing algorithms for the magnetometer, motion sensor, and microphone (acoustic sensor).

We call the sensor reading at a specific time on a specific sensor on a specific node a *sample point*. When a sensor network starts operation, each sensor on each node in the network produces a sequence of sample points. All the sample points produced by the network form a set and we call it the *global sample set*.

The global sample set is the complete information about what happens in the network. If all the nodes report their sample points to a base station, the base station can collect the global sample set and perform computation with it. This solution has been successfully used in a number of WSNs. For surveillance WSNs, however, this is often not feasible because it is too expensive to collect the global sample set in a sensor network. As an example, a 150-node habitat monitoring WSN, presented in [21], collected temperature, humidity, and barometric pressure sensor readings and routed them back to base stations for analysis. During its 115 days of operation, the network collected and routed 650,000 observations. In VigilNet, the data for a one-minute target detection and classification event, with 200 nodes and acoustic processing, well exceeds 1,000,000 observations. If targets enter the network once a day and we routed all the data (the global sample set) back to the base mote, the system could hardly last a week. Hence, the “sense-store-send” style processing is not suitable for latency-sensitive surveillance systems that require a high sampling rate.

On the other hand, the sequence of sample points on a single node does not have enough information to support reliable detection and classification. As an example, a transient disturbance (such as a curious bird landing on the sensor) may shake the node and trigger PIR and magnetic detections. Individual sensor nodes cannot distinguish such an unexpected event from a moving person with ferrous objects. Generally, observations on an individual node are not a reliable indication of events in a network.

Hence, we must design the detection and classification system so that the sensing and classification functions are reasonably distributed in the network and the sensor nodes can cooperate to detect target signatures, reduce false positives, and achieve reliable and timely classification at reasonable energy cost. This motivates us to choose a hierarchical architecture for the classification system. In fact, the concept of hierarchical processing is not new in WSNs. The unique characteristics of our hierarchical design are in the organization of various components and the distribution of the detection and classification tasks in such a hierarchy so that the system accomplishes the required performance with minimal overhead. Illustrated in Fig. 3, the hierarchical classification architecture is comprised of four tiers – sensor-level, node-level, group-level, and base-level. The classification result is represented by a data structure called the confidence vector. The confidence vector comprises the confidence levels for the corresponding targets, and is used as a common data structure to transport information between different levels of the classification hierarchy.

The lowest level deals with individual sensors and comprises the sensing algorithms for the corresponding sensors. With communication being a costly operation, the sensing algorithms need to perform local detection and classification as much as possible. After processing the sensor data, each sensing algorithm delivers the confidence vector to the higher level module – the node-level detection and classification module.

The node-level classification deals with output from multiple sensors on the node. The fusion of the data from various sensors exposes more useful information than can be obtained from any individual sensor. Hence, the node-level sensing algorithm must correlate the sensor data from individual sensors and form node-level classification results. Such a correlation can enhance the detection and classification accuracy on individual nodes – different sensors may strengthen the confidence of each other’s classification results and invalidate false positives. Furthermore, the node-level classification module monitors the sensors’ status and performs sanity control over sensors. For example, it detects and shuts down faulty sensors. Though these functions are all important aspects in the hierarchical classification architecture, this paper does not detail the design of the node-level classification because, compared to other components, it is not the challenging part in the system.

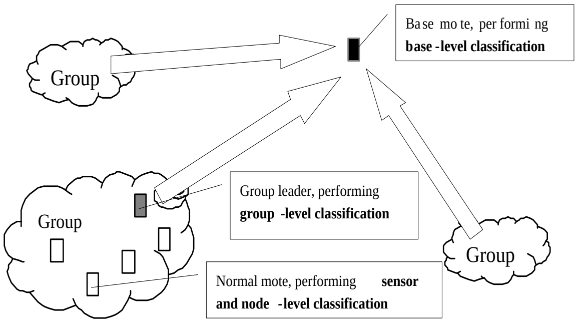

Base mote, performing
base-level classification
Group
Group leader, performing
group-level classification
Group
Group
Normal mote, performing sensor
and node-level classification

Figure 3: Hierarchical Classification Architecture

The sensor-level and node-level classification functions both reside on a single node. The level above, the group-level classification, is performed by groups of nodes. Such groups are managed by a middleware called EnviroSuite [15], which provides a set of distributed group management protocols to dynamically organize nodes in the vicinity of targets into groups and elect leaders among them. These leaders are designated to collect the node-level classification results from individual members and, based on them, perform the group-level classification. Thus, the input to the group-level classification is the node-level confidence vectors rather than a bulk of sample points. This greatly reduces the volume of information transmitted between group leaders and members. Group leaders have much better views of targets compared with individual nodes. Therefore, besides group-level classification they are able to execute more complicated tasks which are extremely hard or even impossible for the node-level. Examples include suspicious report/node detection (based on spacial and temporal correlations among members) and aggregate attribute computation (e.g., computing average member locations as estimates of target positions).

The highest level in the hierarchical classification architecture is the base-level classification. The group-level classification results are transported via multiple hops to the base mote, serving as the input to the base-level classification algorithm. The base-level classification algorithm finalizes the sensing and classification result, as well as computing attributes (e.g., velocity) of the event.

In the following subsections, we present sensing algorithms for the magnetometer, the motion sensor, and the acoustic sensor, focusing on their unique characteristics. Some techniques are used in more than one sensing algorithm. To avoid redundancy, we present them in the sensing algorithm where their purpose and effects can be most clearly explained. In Section 4.4 and 4.5, we describe the group-level and base-level classification, respectively.

4.1 Sensing Algorithm for Magnetometer

In VigilNet, the requirement for magnetic sensing is to detect vehicles and persons with ferrous objects. Since the magnetometer circuitry in the ExScal mote senses a wide range of signals with a fine granularity, we can use it to measure deflection of the magnetic field caused by motion of ferrous objects (e.g., vehicles or weapons). Straightforward as it looks, challenges abound in designing a reliable magnetic sensing algorithm for the low-power sensor network platform.

First, raw ADC readings easily saturate due to the aforementioned granularity/range problem. The ADC on the ExScal platform is 10 bits wide, representing 1024 values. But the wide range of signal intensity combined with a fine granularity requires a much larger value set than the available 0 to 1023. Second, the response latency is too long for accurate signal waveform extraction. The magnetometer circuitry needs about 40 milliseconds to stabilize, and each tuning of the potentiometer needs about 50 milliseconds to stabilize. Third, electromagnetic noise from the circuit board lowers the S/N ratio and imposes serious problems on the magnetic sensing algorithm to distinguish signals from noise. Fourth, thermal drift is a severe issue – When the ambient temperature changes, the sensor readings change accordingly. Finally, radio transmission interferes with the magnetometer sensing circuit.

Among the five issues, the response time is a hardware characteristic. We cannot eliminate such delays. Instead, we measure such delays and reduce them to as low as safety allows. The radio/magnetometer interference can be solved by scheduling the radio and magnetometer to work in separate time slots. The other three issues are more interesting research questions with practical importance to a number of amplitude based sensors. Hence, Section 4.1.1 discusses the sensor reading and signal/noise ratio, and Section 4.1.2 discusses how to deal with the thermal drift. We also use the magnetic sensing algorithm as an example to discuss the trade-off between sensitivity and resilience in Section 4.1.3.

4.1.1 Mag-Points

As mentioned above, raw ADC readings are not suitable to represent the magnetic field intensity, and the magnetometer suffers from a low signal/noise ratio. Both issues relate to a basic question: how to provide credible sensor readings with semantics that higher-level signal processing algorithms can easily use? Hence, we handle these two issues together, by transforming the the raw sensor reading into a 32-bit uniform measure, the Mag-Point.

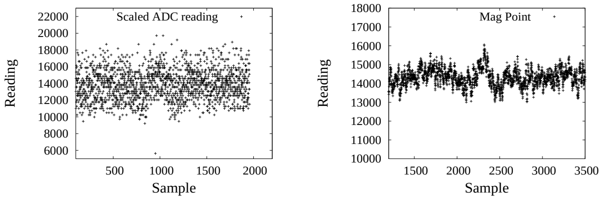
First, the sensing algorithm transforms the raw ADC reading into a scaled ADC reading. The numeric value of the raw ADC reading (r) is determined by the voltage across the magnetic signal line and a reference line. The voltage on the reference line is determined by a digital potentiometer setting. By studying the relation of the changes of potentiometer value (p) with the changes of ADC reading, we map the potentiometer value into certain ADC units. On ExScal motes, experiments reveal that 1 unit of potentiometer change equals 210 ADC units. At run time, as the magnetic signal varies, the sensing algorithm dynamically searches and sets the potentiometer to adjust the reference voltage to a suitable level, and combines r and p to acquire scaled ADC readings (s) using the linear formula: $s = 210 \cdot p + r$

Then, the sensing algorithm averages scaled ADC readings to acquire Mag-Points, using the following moving average

$$\begin{aligned} &\mathfrak{m}_0 = s_0 \\ &\mathfrak{m}_n = \alpha_{mp} \cdot s_n + (1 - \alpha_{mp})\mathfrak{m}_{n-1} \end{aligned}$$

Here $\mathfrak{m}_n$ is the n^{th} Mag-Point, and s_n is the n^{th} scaled ADC reading. The process of generating Mag-Points from raw magnetometer signals filters out high frequency noise and the results are relatively reliable measures representing the current magnetic field intensity. As a comparison, Fig. 4(a) shows the waveform of raw magnetic signals (scaled ADC readings) sampled at 32Hz when an iron bar moved at 5 feet away. As we can see, the signals of the moving iron bar are hidden in high noise.

In contrast, Fig. 4(b) shows the waveform of the Mag-Points, with $\alpha_{mp} = 1/18$, for the same target. The signal is more evident with Mag-Points, which filters out a large part of the noise. Therefore, the Mag-Point is not only a uniform numeric value that is easy to use, but also a loyal indication of the magnetic field intensity that higher-level algorithms can rely on. Such a low-complexity technique is applicable to many amplitude based signals.



(a) Scaled ADC readings (b) Mag-Point readings

Figure 4: Scaled ADC readings and Mag-Points collected from the same sensor in two consecutive runs

4.1.2 Thermal Drift

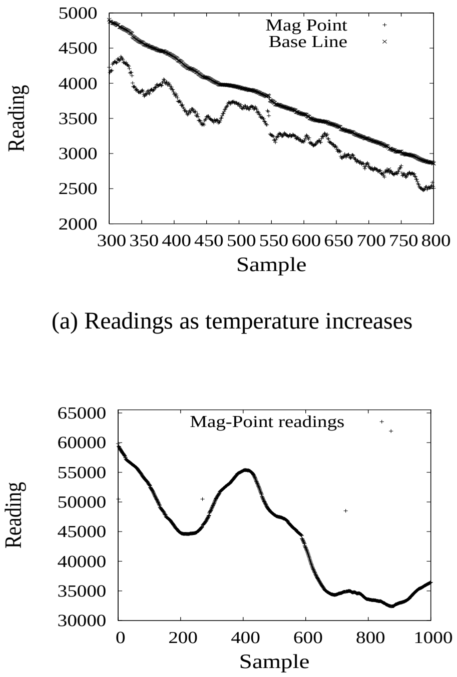
Thermal drift is the most difficult noise the sensing algorithm needs to filter out. Fig. 5(a) shows the magnetometer observations on the X-axis when a sensor node was moved from an air-conditioned room to outdoors on a sunny day. The Mag-Point readings, sampled at 32Hz, fluctuated and dropped quickly in about 15 seconds. Sometimes, the thermal drift is identical to a ferrous target. Fig. 5(b) shows readings collected at noon on a cloudy and windy day. The sensor node was an ExScal mote version 1, which has no enclosure. The frequent alternations of sunshine and shadow caused the temperature of the exposed magnetometer to change quickly. Note that the readings from 300 to 500 (about 6 seconds) is similar to a car moving slowly. Such an intrinsically ambiguous thermal drift cannot be filtered out algorithmically. In such situations, other measures must be employed to avoid such ambiguity. Packaging is the most important supplementary factor that ensures that thermal drift does not produce ambiguous signal waveforms.

Assuming intrinsically ambiguous thermal drifts are eliminated by methods other than software, frequency based analysis can be used to filter out other thermal drifts. The thermal drift is a relatively slow change, i.e., low frequency noise. To eliminate this, the sensing algorithm uses another moving average, which assigns more weight on history, to compute the current base signal line. The formula for B_n (the n_{th} point in the base signal line) is

$$\begin{aligned} &B_0 = s_0 \\ &B_n = \alpha_B \cdot s_n + (1 - \alpha_B) \cdot B_{n-1} \end{aligned}$$

Fig. 5(a) also shows the base signal line. As we may notice, when the sensor readings change, the base signal line readings change at a slower speed than the Mag-Points. When the Mag-Points deviate from the base signal line for an amount larger than a threshold, detection occurs.

By using two moving averages with very low computational complexity, the magnetic sensing algorithm filters out both high frequency and low frequency noise, solves the problems of non-uniform



(a) Readings as temperature increases

(b) Readings on a cloudy and windy day

Figure 5: The impact of temperature on the magnetometer

sensor reading, low signal/noise ratio, and thermal drift, and accomplishes a resilient detection algorithm.

4.1.3 Trade-off between sensitivity and resilience

The parameters α_{mp} and α_B affect the performance of the magnetic sensing and must be carefully chosen so that the magnetic sensing algorithm is not only sensitive, but also resilient to noise and environmental changes.

The parameter α_{mp} affects how effectively the algorithm averages out high frequency noise. If $\alpha_{mp} = 1$, there is no noise filtering. As we decrease α_{mp}, high frequency noise is filtered out by the averaging process, and small signals are able to emerge from the background noise. When α_{mp} approaches 0, however, the history readings overwhelm the new reading so much that signals lasting for a short period of time cannot distinguishably change the Mag-Point readings. Hence, the algorithm becomes unable to detect a target unless it is moving very slowly. This means that the sensitivity decreases, and the false negative rate increases, when α_{mp} is too large, or too small.

The parameter α_B aims to establish a baseline to characterize the ambient magnetic field strength without targets. If $\alpha_B = \alpha_{mp}$, the baseline reading B_n is the same as the Mag-Point $\mathfrak{m}_n$, and there can be no detection since their difference is always 0. With α_{mp} fixed and α_B decreasing, the baseline becomes more stable. When a target approaches, the Mag-Points change faster than the baseline. Generally, the smaller the α_B, the larger the difference between the baseline and Mag-Points, and the more sensitive the magnetic algorithm is. However, when α_B increases, the baseline also becomes less adaptive to environmental changes, such as the temperature change, and becomes more likely to report false positives. When $\alpha_B = 1$, the algorithm has the maximum sensitivity, but shows a very weak resilience because it does not adapt to the environment at all and thus any environmental change can trigger a false positive.

As we can see from the analysis above, choosing a suitable α_{mp} and α_B is a design decision that affects the magnetic sensing algorithm's performance. Their ranges of suitable values are dependent on the application requirements, the sensor properties, and the expected environmental variability. In VigilNet, we choose $\alpha_{mp} = 1/4$ and $\alpha_B = 1/64$, after weighing the above factors and experimenting with a number of settings.

4.2 Sensing Algorithm for Motion Sensors

The task for motion sensors is to detect movement of an object in the region where the sensor network is deployed. The motion sensors on sensor boards are peroelectric infra-red (PIR) sensors. They sense changes in the thermal field over the region. During the time when an object is moving through, the variations of the thermal field result in unbalanced infra-red signals detected by the lens pairs in the PIR sensor, leading to positive detections. Unlike the magnetometer, the PIR signals are AC signals, not amplitude based. A distinctive challenge to designing a reliable motion sensing algorithm is the weather. Hence, we introduce a motion sensing algorithm, focusing on its low-complexity frequency based processing and environmental resilience.

4.2.1 Increasing S/N ratio by filters

In outdoor environments, the performance of PIR sensors depends heavily on the weather conditions, including wind, temperature and humidity. Wind makes the air move and grass and trees swing, causing the thermal field to change since the air temperature is not uniform and grass and trees have different temperatures. Fig. 6(a) shows PIR data collected by a sensor in grass on a hot, humid and windy day and Fig. 6(b) is the spectrum of the signal. There is a moving target in the area between 60s and 70s. On hot, humid and windy days, when the sensors are placed in grass, a simple energy detector either generates false positives, if using a low threshold, or misses targets, if using a high threshold. We observe that the low frequency component, less than 1 Hz, dominates the noise. When a target moves through, the frequency components larger than 2Hz become significant. This motivates us to explore frequency based signal processing on PIR data.

Because of the limited computation resource and the time constraints of the application, we design a high pass filter as follows.

$$\begin{aligned}\mathfrak{m}_0 &= 0\\ \mathfrak{m}_n &= s_n - s_{n-1} + 0.9\mathfrak{m}_{n-1}\end{aligned}$$

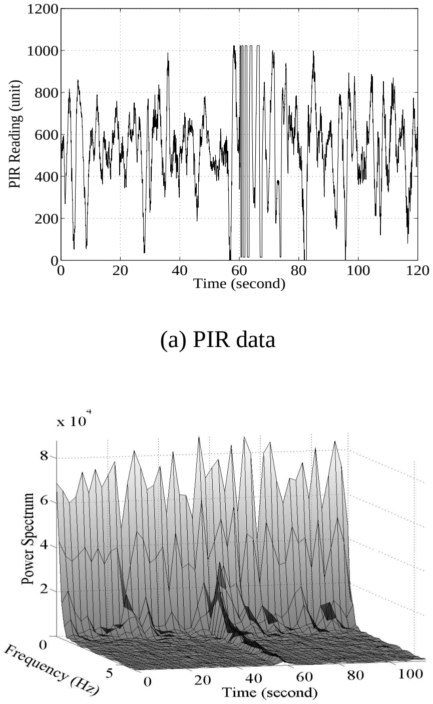
Fig. 7(a) shows the frequency response of the filter. Fig. 7(b) shows the spectrum of filtered PIR data on a hot and windy day collected by a sensor in grass. The coefficient 0.9 is decided empirically with different filters on PIR data collected outdoors. Although a higher order filter could achieve lower gain for components less than 1 Hz, it does not significantly improve the performance.

Fig. 8(a) is the filtered signal of the one in Fig. 6(a). When the moving object passes, there is considerable energy variation in the signal. A simple energy detector can then be applied to the filtered signal to detect movements with a low false positive rate.

4.2.2 Unsupervised adaptation to environment

In the motion sensing algorithm, the energy based target detection threshold must be set based on the noise level. However, in realistic environments, the noise level is not fixed. The low frequency noise is very weak on cold and arid days, but can be strong on hot and windy days. Fig. 8(a) and Fig. 8(b) compare the PIR data for two different scenarios. Obviously, we cannot achieve good performance with a fixed threshold in all types of weather conditions.

To solve this problem, we use an unsupervised adaptation tech-



(a) PIR data

(b) Spectrum of PIR data

Figure 6: PIR readings from sensor in grass

nique to adjust the threshold. The sensors continuously compute the noise level based on local measurements and adapt the threshold proportional to the noise level. To compute the noise level, the motion sensing algorithm monitors the maximum power p_n of the filtered signal within a time window. The noise level ϵ_n is updated by the following computation.

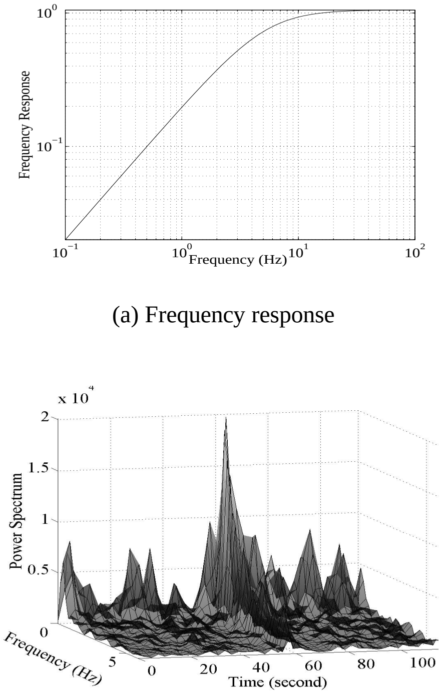
$$\epsilon_n = \begin{cases} p_0 & n = 0 \\ 0.98\epsilon_{n-1} + 0.02p_n & \epsilon_{n-1} < p_n \\ 0.75\epsilon_{n-1} + 0.25p_n & \epsilon_{n-1} \geq p_n \end{cases}$$

The motivation of this formula is to let ϵ_n increase and decrease at different speeds. This is because the weather changes slowly therefore we don't need to increase the noise level quickly to adapt to the weather change. A small weight on p_n for $p_n > \epsilon_{n-1}$ avoids the identified noise level increasing too fast when there are moving targets. Once there is no target, we decrease the noise level quickly with large weight on p_n for $p_n \leq \epsilon_{n-1}$. Fig. 9(a) shows the signal power of filtered PIR data in Fig. 8(a). The dashed curve is the identified noise level and the dashed-dotted curve is the updated threshold that is $1.5\epsilon_n$. An exceptionally large noise after 80 seconds causes a false detection.

The motion sensing algorithm monitors the number of detections within a time window and defines the percentage of the detection within the time window to be the confidence of a target in the field. Fig. 9(b) shows the sensor confidence for the signal in Fig. 6(a).

4.3 Sensing Algorithm for Microphone

VigilNet uses acoustic sensing to differentiate between vehicles and humans. Acoustic sensing is unique in its relatively high frequency in sampling and processing. The resource constraints make it challenging to design a reliable acoustic sensing algorithm. First, the simultaneous use of magnetic and motion sensing limits the



(a) Frequency response

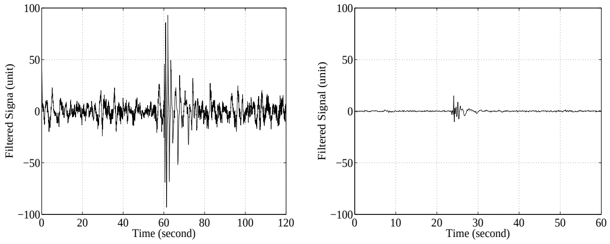
(b) Spectrum(filtered data)

Figure 7: PIR data filter

rate at which we can collect acoustic samples. Second, the CPU must remain available at all times to process incoming messages. Third, the system must continuously process acoustic data in order to detect and identify targets in time. Fourth, our whole surveillance system only has 4KB RAM for its functioning: our acoustic algorithm should occupy as little memory as possible.

Frequency analysis could be an effective method to conduct acoustic detection and classification. Unfortunately, computing the frequency spectrum by FFT and analyzing the spectrum are expensive operations in our design context. The number of multiplications it takes to get the frequency domain results is $\Theta(N \log_2 N)$. The microcontroller ATmega128L used in the Mica2 and ExScal motes does not support native floating-point multiplication and the clock rate is between 4 MHz to 8 MHz. Xu shows that in [23], it takes a Mica mote with a 4MHz processor 30 seconds to finish a 512-point FFT. Hence, an ExScal mote with a similar processor runs 15 seconds for a 512-point FFT even if it is running at its maximum 8 MHz clock rate. Such a long latency is not acceptable in our application. The space complexity is another issue. Although there are in-place fixed-point FFT solutions, even when we consider a 1024-point FFT and each data point is 16 bit, an in-place solution still uses at least 2 KB space just for the data points. In order to save the 16-bit trigonometric value table, which is necessary for FFT calculation, another 2 KB is needed. In Mica/Mica2 series motes, the RAM size is 4KB and a large proportion of the RAM needs to be assigned to other modules. Of course, the off-chip Flash can be used as secondary storage, but frequent writes to the Flash makes the FFT computation even slower and quickly damage the Flash.

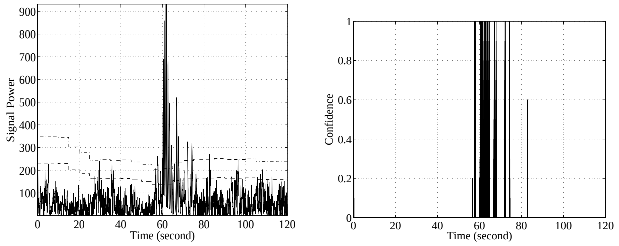
We, therefore, choose a less costly power-based scheme. Each time we obtain a new acoustic sample, we update an exponentially



(a) Data from a sensor in grass(hot and windy)

(b) Data from a sensor on ground(cold day, no wind)

Figure 8: Filtered PIR readings



(a) Signal power and noise level

(b) Sensor confidence

Figure 9: Signal power and detection confidence

weighted moving average of acoustic sample values, noted m_1:

$$\begin{aligned} &m_{1,0} = s_0 \\ &m_{1,t} = \alpha_1 \cdot s_t + (1 - \alpha_1) \cdot m_{1,t-1} \end{aligned} \qquad (1)$$

Where $m_{1,t}$ is the current value of m_1, $m_{1,t-1}$ is the previous value of m_1, s_t is the current microphone reading and α_1 is a constant determining the relative importance of recent readings. In our current system, α_1 is empirically determined to be 0.001. Fig. 10 graphs the raw acoustic data provided by an ExScal mote when three vehicles pass. The corresponding evolution of $m_{1,t}$ is also presented. We use this moving average to serve as a reference in the computation of E_t, a variable related to the instantaneous acoustic energy:

$$E_t = |s_t - m_{1,t}| \qquad (2)$$

Then we compute an auto-adapting acoustic threshold that detects acoustic events. We chose this threshold to be the sum of an exponentially weighted moving average of E_t, noted m_2, plus what we name an exponentially weighted moving standard deviation, noted d_2. equations:

$$\begin{aligned} &m_{2,t} = \alpha_2 \cdot E_t + (1 - \alpha_2) \cdot m_{2,t-1} \\ &v_{2,t} = \alpha_2 \cdot (E_t - v_{2,t})^2 + (1 - \alpha_2) \cdot v_{2,t-1} \\ &d_{2,t} = \sqrt{v_{2,t}} \end{aligned} \qquad (3)$$

Here v_2 represents an exponentially weighted moving variance. In our current implementation, α_2 is empirically determined to be 0.02. When $E_t > m_{2,t} + d_{2,t}$, the algorithm considers that the acoustic threshold has been crossed.

A circular table maintains the number N_t of acoustic threshold crossings that occurred during the last 1280 milliseconds. The value of N_t determines the nature of an acoustic event. When N_t is greater than a certain predetermined value T ($T = 8$ in our experiments), the system signals the detection of a vehicle. Fig. 11 graphs the values of E_t, $m_{2,t}$ and $d_{2,t}$ for the previously mentioned data set for three passing vehicles. Fig. 12 represents the corresponding values of N_t. We remark that the mote triggers a car detection event during the first three seconds of the algorithm execution. This erroneous detection is due to the fact that, at the beginning of the algorithm execution, the moving averages are arbitrarily set to zero. To resolve this problem, the system disregards acoustic detection results during the first five seconds of its execution.

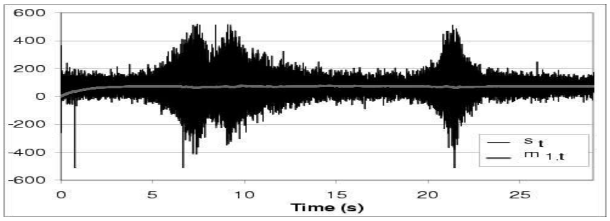


Figure 10: Raw acoustic data from three vehicles

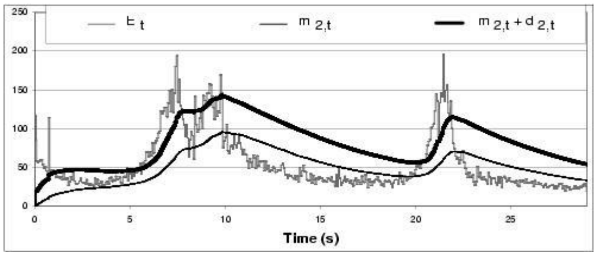


Figure 11: Acoustic Energy and threshold for three vehicles

4.4 Group-Level Classification

Distinguished from previous in-network data processing schemes [6, 9, 16], the groups in the VigilNet are more dynamic in the sense that they are formed in response to an external "event", which corresponds to a target in VigilNet, and migrate with the movement of the event. The details of the group forming, migration, and deletion can be found in [15]. In this section, we introduce groups' functions in the classification system – collecting, filtering, and aggregating node-level classification results, as well as triangulating the estimated target locations.

Each group has a statically assigned group leader. When events occur, group members periodically report to the group leader. The reports usually consist of node information (e.g., node ID and location), group information (e.g., leader ID and group ID) and event information (e.g., confidence vectors). The group leader aggregates the confidence vectors from group members, computes group-level confidence vectors and reports them to the base-level classification module via multi-hop communication. This scheme greatly reduces the amount of network traffic and, consequently, the energy consumption of the WSN.

Data aggregation contains several tunable parameters that affect different aspects of its performance. One parameter, minimum degree of aggregation (MDOA), defines the minimum number of distinct reports required to form a valid group-level confidence vector. An adequately high MDOA value enhances the credibility of group-level classification results. Hence, it is an important system parameter and has an impact on the performance of the detection and classification, which is to be discussed in Section 5.2.

In the classification system, the groups have a one-to-one mapping to physical events. An implicit assumption is that events always keep a far enough distance among them, so that membership

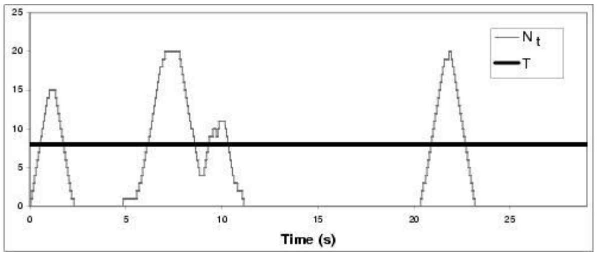


Figure 12: Number of threshold crossings for three vehicles

of nodes to the corresponding groups can be determined without ambiguity based on spatial adjacency to one of the events. This results in a limitation on detecting multiple simultaneous targets – for events that become close enough or cross each other, if they share the same sensory signature (e.g., two persons walking together), the current classification system cannot separate them.

If events are with different sensory signatures, different classes of events can be resolved based on history data after events deviate from each other. However, before such events deviate, there is still a temporary ambiguity. For instance, when a group for a vehicle and a group for a person cross, the person triggers detections on the motion sensors, and the vehicle triggers detections on the motion, acoustic, and magnetic sensors. Hence, the two groups merge to be a group for a vehicle, sensing an event with motion, magnetic, and acoustic features. Later, when the person and vehicle deviate, the ambiguity will be resolved and two groups will be formed. This is another limitation of the current classification system – events of different signatures may still have "temporary ambiguity" because the groups are formed by detecting nodes, not by nodes detecting a specific type of signature.

Potentially, temporary ambiguity can be resolved by group management with a finer granularity – a group is formed for a specific type of signature, hence multiple groups co-exist in the same vicinity. Another solution is to have the base mote disambiguate the events, based on track history and assumptions on trajectory. However, our current system does not pursue either of the approaches in order to keep time, space, and communication complexity low. Instead, we design the system so that the effect of the ambiguity is minimized. Specifically, a group still reports an event even when there is temporary ambiguity, allowing the system to still reacts to the event. Furthermore, when there is ambiguity, the classification tends to categorize it as a class of a higher alert level (e.g., a person and a vehicle are identified as a vehicle). On the other hand, for applications that require a better disambiguity capability, the hierarchical classification architecture allows more sophisticated group-level and base-level algorithms to be incorporated.

4.5 Base-Level Classification

The highest level detection and classification are conducted on the base mote. It takes the group-level classification results as input and computes the final classification results. Since the base mote has a global view of the classification process, it conducts the tasks requiring global knowledge, which is not available to individual nodes or groups. In order to further reduce false positives, spatial and temporal correlations among the tracking reports must be leveraged. Intuitively, the base mote deems that two reports in a certain time frame are from the same target if their locations are close. The base mote keeps a history of recently received reports. With the history of reports, the classification and velocity calculation of each target can be accomplished with high accuracy.

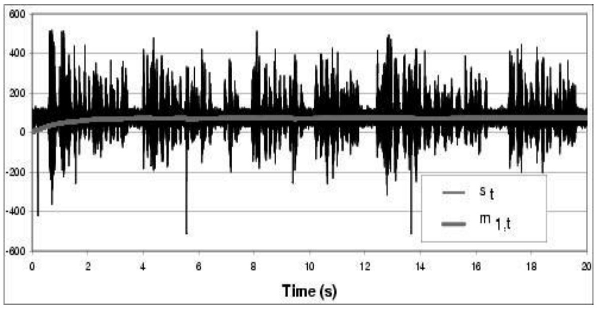


Figure 13: Raw acoustic data from human speaking

In the RAM of the base mote, a small data structure for each target is maintained. The data structure includes the recent location of the target, the latest timestamp, accumulated sensor values and a pointer to the information of the last report for the target. The base mote chooses the target whose recent location is the closest to the location of the incoming report and decides that the report belongs to the target. If there is no target or the closest distance from the recent location of any target to the location of the reports is greater than a predefined threshold, the report is considered to be from a new target. This threshold needs to be tuned in real-system testing. If it is too large, reports from multiple targets may be categorized into one group. If it is too small, two consecutive reports from a single target may be categorized into two groups. Currently in our system we use a threshold of 60 meters, which shows good performance results in experiments. In order to minimize the number of false positives, a target is reported to the front end interface only if the number of reports for it exceeds a predefined threshold. With this approach, most sporadic false positives can be filtered out.

Once a target accumulates enough reports, the base mote reads its history and applies a linear regression to calculate the velocity of the target, because velocity is one of the most important aspects for moving target tracking, and it is of great interest to the end users. The least square regression approach has been used in many scientific and engineering fields for a long time and is believed to be highly robust against small numbers of outliers. For each direction, the timestamps and the coordinates of the locations of the the most recent reports are used in the regression. The least square algorithm gives the average changing rate of the coordinates over time. This rate serves as the component of velocity along the direction. With the information of both velocity components, we can get the velocity of the target including the knowledge of its moving direction.

5. SYSTEM PERFORMANCE

In this section, we evaluate the the performance of VigilNet with a focus on the detection and classification performance. First, we evaluate the performance of sensing algorithms in Section 5.1. Then, Section 5.2 studies the group-level classification by analyzing the impact of MDOA on the classification performance. Finally, Section 5.3 assesses the overall system performance.

5.1 Evaluation of sensing algorithms

Among the three sensing algorithms, the acoustic sensing algorithm has the highest sampling rate and CPU utilization. Since a detailed analysis of all three algorithms has to be lengthy, we choose to study the acoustic sensing algorithm as a representative and evaluate its detection rate. To evaluate the performance of the acoustic sensing algorithm, we deploy 7 sensor nodes in a line with 3-meter spacing. This line is perpendicular to the trajectory of a

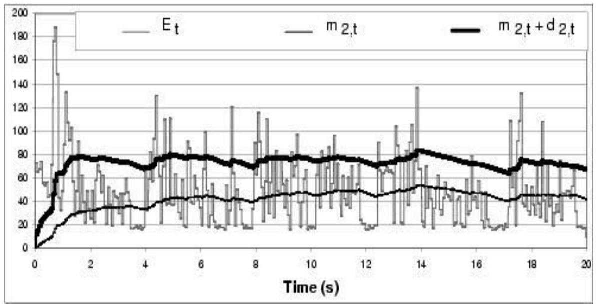


Figure 14: Energy and auto-adaptive threshold for human speaking

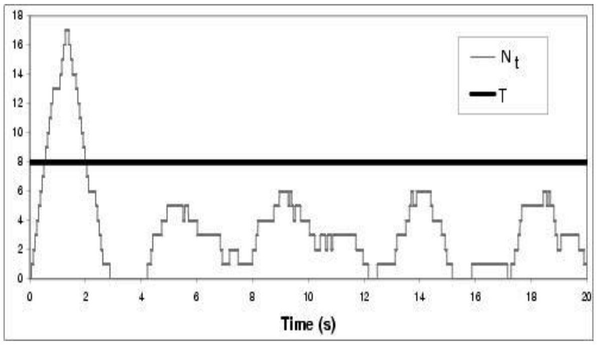


Figure 15: Number of threshold crossings for human speaking

passing car, with the first node located 3 meters from the trajectory. We drive the car at three different speeds: 10, 15 and 20 miles per hour. We realize ten trials for each speed in a parking lot and compute the success rate of our algorithm at various distances and speeds. Fig. 16 presents the results of this experiment. We observe that the success rate of our algorithm decreases as the distance to the car increases. Also, the algorithm is more successful when the car moves at higher speeds: this is not surprising as a rapidly moving car generates more acoustic power. In VigilNet, sensor nodes are approximately 33 feet (10 meters) away from each other. A sensing range of 16.5 feet (5 meters) guarantees the detection of a target traversing the field. Considering the resource constraints, our design does not emphasize very high detection rate on individual nodes. Hence, the performance of the acoustic sensing, with a detection rate of 90% at 30 feet (9 meters), is sufficiently good.

To demonstrate how our acoustic algorithm reacts to other sound sources, we experiment with a human speaking loudly at a distance of 1.8 meters (6 feet) from an ExScal mote. Note that, without sophisticated frequency analysis, the acoustic sensing algorithm is not designed to distinguish human voice and vehicle sound. However, according to experimentation, a human, even if speaking loudly, does not generate as much acoustic energy as a car passing close to the sensor node. This is why the algorithm, evaluating acoustic energy, can differentiate between these two types of targets. On the other hand, if the acoustic sensing algorithm shows a good resilience to human voice, which is strong at such a short distance, it indicates a good performance against background noise. The results are presented in Fig. 13, Fig. 14, and Fig. 15. Clearly, the acoustic algorithm does not trigger any vehicle detection event except during the first three seconds. As we said earlier, acoustic detections during this initial phase are ignored. Hence, from the system's view, no human speaking events are reported as vehicles in this test.

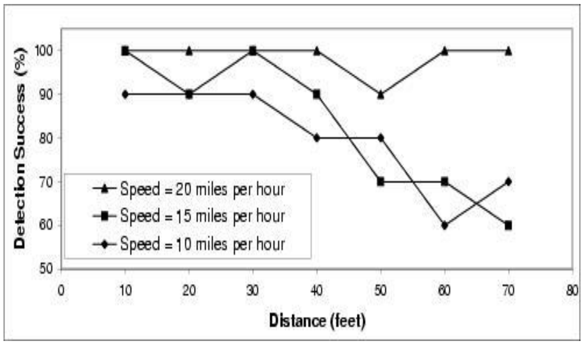


Figure 16: Acoustic detection performance.

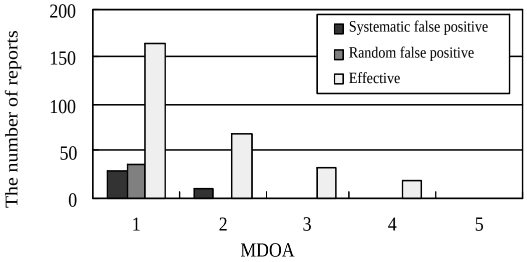


Figure 17: The impact of the MDOA.

5.2 Evaluation of the group level classification

Among the functions of group level classification, the MDOA (the minimum degree of aggregation) is of special importance to the detection and classification performance. It not only controls the aggregation of member reports and reduces the network traffic, but also reduces the false positive rate of the system.

From a system designer's point of view, false positives roughly fall into two categories. One category of false positives are due to unexpected disturbance imposed on a small number of nodes. For example, when a wild animal touches a sensor node, that node may sense motion and magnetic signals and report false positives. A loose connector, a piece of shortcut wire, or a malfunctioning sensor may trigger continuous wrong readings from the sensor. Such false positives are mostly independently and randomly distributed and only occur at a reasonably low rate. We call this category of false positives "random false positives". In some other situations, a large percentage of the sensors in a WSN become much more probable to report false positives, because of flawed design or unexpected disturbances imposed on a large percentage of the network. False positives in this category are usually correlated, and sometimes bursty. For example, when the sensor driver has a bug, or there is a storm, the entire network may report false positives in large quantity. We call this category "systematic false positives".

By using a suitable MDOA, the group level classification can significantly reduce random false positives, and noticeably mitigate the effect of systematic false positives. Fig. 17 shows the result of a test studying the relationship between the MDOA and the number of false positive reports. Performed at an outdoor parking lot with a test VigilNet system of 37 nodes, the test involves the magnetometer and motion (PIR) sensor detecting a moving vehicle. The thresholds for the motion sensing algorithms are set to an extremely low value so that there are frequent false positives. Also, on the current hardware platform, the magnetometer is sensitive to LEDs. Hence, we let the LEDs blink at the beginning of the classification stage so that the magnetometers acquire some wrong data and generate false positives.

The testing procedure is as follows. First, we start the system, with MDOA set to 1 (all reports are delivered to the base mote). At this point, the LED blinks and the initial low threshold triggers false positives on the magnetometers and motion sensors. We record the number of reports in the first 32 seconds, and use them, as a reasonable approximation, as the number of systematic false positives. We record the number of reports in the following 3 minutes as the number of random false positives. Then, we send a middle-sized car to the WSN field, record the number of reports delivered during the tracking process (35 seconds) as the number of effective reports. We also examine whether the classification result is correct. Tests are repeated and statistics are collected for various MDOA settings.

As we can see from Fig. 17, when the MDOA is 1, there are 29 systematic false positives and 36 random positives. Such a high false positive rate confuses the base level classification algorithm and the system reports false targets. On the other hand, the system is still able to track the real target – the 165 effective reports make the system successfully detect and classify the vehicle.

When the MDOA increases past 1, all random false positives are eliminated. And the number of systematic false positives reduces by 85% – from 65 to 10. Meanwhile, the number of effective reports also reduces from 165 to 69. But the system is still able to detect and classify the target vehicle correctly.

When MDOA increases to 3 or 4, all the systematic and random false positives are removed. And we verified that, though the number of effective reports is further reduced, the system detects and classifies the target correctly. When MDOA is 5, no reports are delivered in the system.

In conclusion, adjusting the MDOA is an important method to reduce the number of false positives in a WSN, and significantly enhance it's performance. Meanwhile, a too-high MDOA lowers the system's sensitivity.

5.3 System level performance

In this subsection, we evaluate the VigilNet's performance as a holistic system. Especially, we measure how fast the network classifies targets and how accurately it computes the target's attributes. Among a number of attributes, velocity is our major interest and a good representative of high-level target attributes that cannot be accurately computed on individual nodes. Hence, in this section, the discussion of attribute computation focuses on velocity. We deployed and tested the VigilNet in an airfield in the July and December of 2004. Unless otherwise specified, the performance data in this section is collected from tests on these two deployments.

The test scenario involves moving targets traveling through the network following a straight trajectory. A moving target may be a vehicle, a person, or a person carrying a ferrous object. The network comprises 200 ExScal motes.

Table 1 shows statistics collected in an outdoor test site. Ten targets are tracked in two runs. In this test, the required minimum degree of aggregation is set to one in the group level classification in order to inspect the base mote's ability to filter out false positives. All the 10 targets are detected and correctly classified. In total, the network delivers 441 reports to the base mote, which, after processing these reports, delivers 71 reports to the external device. Interestingly, the network delivers more reports in run 1 than in run 2, even though there are more targets in run 2. The reason is that the number of reports from the network depends not only the number of targets, but also the amount of time the targets stay in the network, the types of the targets, and the number of group leaders in the network. In these two runs, the network generated totally 29 false positive reports at the node level and the group level, but all of them are filtered out by the base mote. Hence, from the system's view, there are no false positives and, since all targets are detected, there are no false negatives. Not surprisingly, the network's performance is better than individual nodes. The reasons is that the natural redundancy in a densely deployed network help reduce the false negative rate at the system level.

Run No.	Run 1	Run 2
Duration(s)	271	758
Group-level reports	261	180
Reports after filtering	29	42
Actual targets	4	6
Correctly classified targets	4	6
False negatives	0	0
Filtered false positives	5	24

Table 1: Statistics of classifying 10 targets in two runs

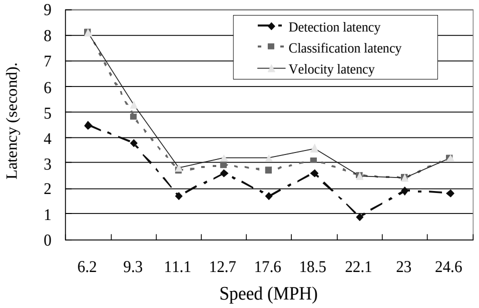


Figure 18: Latencies for vehicles at different speeds

Fig. 18 plots the detection latencies, classification latencies and velocity calculation latencies for vehicles at various speeds. Generally, the detection latency is lower than the classification latency, and the classification latency is lower than the velocity calculation latency. This difference reflects different amounts of information required for the detection, classification, and attribute calculation. The classification of a target employs a longer history of reports than the detection. The velocity calculation needs a even longer history, in order to accomplish a precise linearly-fit inference of the velocity. Also, the latencies reduce when the speed increases. The reason is that, when the target is traveling at a low speed, the time for the target to travel past multiple nodes dominates the total latency. When the speeds increase, the latency remains at a certain level. This is because, when the speed is high, the processing, queuing, and group-level aggregation latency dominate the total latency.

In the runs shown in Fig. 18, all targets are classified correctly, indicating a satisfactory classification capability. Meanwhile, it is interesting to compare the calculated velocity with the velocity shown on the vehicle speedometer. Our record shows that the range of error between the calculated velocity and the real velocity ranges from -7.5% to $+15\%$.

The motion of persons and persons with ferrous objects have are similar characteristics because the moving carriers are of the same type – human beings. Hence, they show similar latencies in the tests. However, their latencies are longer than those for vehicles. Fig. 19 shows the average detection, classification and velocity

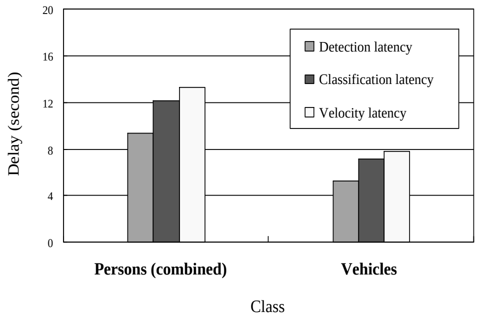


Figure 19: Average latencies for persons and vehicles

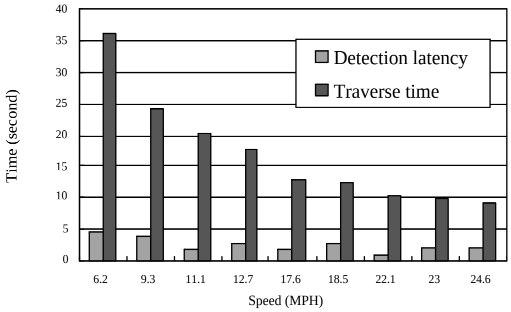


Figure 20: Detection latencies and traverse time for vehicles

calculation latencies for the two classes of persons combined, and vehicles. We notice detection latency for persons combined is 77% longer than that for vehicles. This is because the persons travels much slower in the network than vehicles, hence it takes longer time for persons to hit enough sensor nodes and trigger enough detection and classification reports to establish a sufficient confidence level for detection on the base mote.

As mentioned in Section 1, a surveillance WSN must operate in a timely manner. Given that the detection latencies for persons and some slow vehicles can approach 10 seconds, it is necessary to verify that the latencies are all within an acceptable range. One design detail in VigilNet is that the deployment ensures that a target has to travel about 330 feet (100 meters) to traverse the network. Based on this we can calculate the minimum amount of time that it takes a target to traverse the network, henceforth called the “traverse time”. Suppose a person travels at 2–10MPH, it takes 22–112 seconds to traverse the network. As shown in Fig. 19, the detection latencies for persons are much shorter than the traverse time. As for vehicles, the traverse times are shorter for faster vehicles, but the detection latencies are usually shorter, too. To examine the timeliness of the detections, we plot the detection latencies and the traverse times at different speeds in Fig. 20. As we can see, the detection latencies are much shorter than the traverse times at various speeds.

From the analysis of performance at the sensor level, the group level, and the base level, we can clearly see the refining process of the detection and classification results. The detection rate at the sensor level is often not perfect. For instance, the acoustic sensing algorithm’s detection rate is about 90% at a distance of 9 meters (30 feet). However, the redundancy of the network nodes ensures that the holistic system has a high detection rate (low false negative rate). The group-level classification significantly reduces the false positive rate and minimizes the network traffic. Finally, the base-level classification refines the detection and classification results by analyzing tracking reports from multiple groups. In summary, the evaluation shows that VigilNet accomplishes an excellent performance in reliable sensing and classification, accurate attribute (velocity) computation, resilient operation in realistic environments, and timely information delivery.

6. METHODOLOGICAL DISCUSSIONS

We believe that the design, implementation and evaluation of the sensing subsystem and classification algorithms should evolve from an ad hoc “art” to established methodologies. Though our experience with VigilNet and a study of several recent surveillance WSNs are not sufficient to abstract such a methodology, the challenges we faced do represent a series of common issues and the design choices we made reflect the diligent thoughts, careful trade-offs, and realistic concerns involved in constructing a realistic, sophisticated, and evolving system. Hence, we conceive that it is valuable to share our view and conception on how to design a detection and classification system, and believe this can help current and future WSN research to establish a systematic methodology for designing such WSNs. For this purpose, we abstract some general guidelines for the development of sensing and classification systems. Most of them are not new concepts, but are of critical importance to the success of realistic systems.

- *Mechanical design.* Though programmers traditionally do not care about the mechanical details of a computer system, designers of a sensor network must pay careful attention to it. For example, without a suitable enclosure, the magnetometer would suffer degraded performance at sudden temperature changes. Generally, the enclosure design for WSN nodes should provide a suitable operating environment to the hardware components [8], besides protecting the node hardware from harsh environmental conditions. Specifically, the positioning and wiring of various components should avoid interference from each other and maximize sensors’ capability.
- *Autonomous operation.* It is infeasible to individually manage the network nodes in a large-scale WSN. Hence, each must operate in an autonomous manner. Specifically, a node must identify, calibrate, and operate its sensors automatically.
- *Fault tolerance.* For a large system to operate for a long period of time on a rough terrain, it must expect the unexpected. For example, strong wind or wild animals may disturb the sensor nodes, displace them, or even destroy them. Also, the large size of the network makes faults a common phenomenon – if each node has a 0.001 possibility to have a hardware fault, a network of 200 such nodes has a 0.18 possibility to contain a faulty node. Though it is infeasible to analyze and intelligently handle all possible situations, the design of such WSNs must, at minimum, deal with failures at various levels.
- *Adaptivity.* WSNs, especially when they are deployed outdoors, show a high level of dynamics. The quality of communication links, the electric characteristics of sensors, and the topology of the network, may continuously change due to internal and external conditions. Hence, many system parameters need to continuously adapt to changes inside and outside the network.
- *Redundancy and collaboration.* The performance of a network of energy-and-cost-effective nodes largely relies on how the nodes collaborate with each other. Enhancing the performance and capability of individual nodes is important. Many developers, including the authors, feel it intellectually exciting to answer the challenge of high-quality signal processing with low-end hardware by novel and prudent architectural and algorithmic designs of individual sensor nodes. Meaningful and important as it is, such an effort, if overemphasized, may prove inefficient or, in some situations, even hazardous, in realistic settings.

For example, an intrinsically difficult trade-off is that the more sensitive the sensing algorithms are, the more vulnerable to the changing environments they become. As another approach, we may choose to make the sensing algorithms less sensitive, but more resilient to environmental changes, and take advantage of the density of the network nodes to enhance the overall sensitivity of the system. Also, as we discussed in Section 5.2, a group of sensor nodes can collaborate with each other to reduce the false positive rates. Such a redundant and collaboration based approach proves to be highly effective.

7. CONCLUSION

We discussed the sensing subsystem in the VigilNet surveillance system and described how the hierarchical classification architecture enables the system to conduct efficient information processing, including detection and classification, in a large-scale WSN. The hierarchical architecture naturally distributes sensing and computation tasks at different levels of the system so that the sensor network can support high-quality sensing and reliable classification without involving special high-power nodes. With evaluation data collected from field tests in physical environments, the evaluation of VigilNet demonstrates excellent performance on the detection rate, classification result, attribute (velocity) computation accuracy, and timely information delivery.

8. ACKNOWLEDGMENTS

Thanks to Anthony Wood, Shashi Prabh, Prabal Dutta, Cory Sharp, Michael Grimmer, Qiuhua Cao, William Bain, Cole Timothy, Mark Satterfield, and Ed Roach, and many other researchers and staff, whose help makes this work possible.

9. REFERENCES

[1] Exscal web site. *http://www.cast.cse.ohio-state.edu/exscal/index.php?page=main*.

[2] Honeywell magnetometers. *http://www.ssec.honeywell.com/magnetic/*.

[3] Mica2 mote. *http://www.xbow.com/Products/productsdetails.aspx?sid=72*.

[4] Micropower inpulse radar by advantaca. *http://www.advantaca.com/radar.htm*.

[5] VigilNet web site. *http://www.cs.virginia.edu/ control/SOWN/index.html*.

[6] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher. Energy-conserving data placement and asynchronous multicast in wireless sensor networks. In *Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, May 2003.

[7] R. Brooks, P. Ramanathan, and A. Sayeed. Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8):1163–1171, 2003.

[8] 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 Fourth Intl. Conf. on Information Processing in Sensor Networks (IPSN'05)*, 2005.

[9] Z. Feng, S. Jaewon, and R. James. Information-driven dynamic sensor collaboration for target tracking. *IEEE Signal Processing Magazine*, 19(2), March 2002.

[10] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, G. Zhou, J. Hui, and B. Krogh. Vigilnet:an integrated sensor network system for energy-efficient surveillance. *In submission to ACM Transaction on Sensor Networks*, 2004.

[11] T. He, S. Krishnamurthy, J. A. Stankovic, T. F. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An energy-efficient surveillance system using wireless sensor networks. In *Proc. of Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, June 2004.

[12] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. Abdelzaher. Achieving Long-Term Surveillance in VigilNet. In *submission*.

[13] J. Hill and D. Culler. Mica: A wireless platform for deeply embedded networks. In *IEEE Micro*, volume 22, pages 12–24, Nov./Dec. 2002.

[14] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. *Proc. of ASPLOS 2000*, Nov. 2000.

[15] L. Luo, T. F. Abdelzaher, T. He, and J. A. Stankovic. Design and comparison of lightweight group management strategies in envirosuite. In *Distributed Computing in Sensor Systems (DCOSS '05)*, June 2005.

[16] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *Proc. of Operating Systems Design and Implementation*, Dec. 2002.

[17] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *Proc. of the 2nd ACM Intl. Conf. on Embedded Networked Sensor Systems (SenSys'04)*, pages 39–49, New York, NY, USA, 2004.

[18] S. Pattem, S. Poduri, and B. Krishnamachari. Energy-quality tradeoffs for target tracking in wireless sensor networks. In *Proc. of 2nd Intl. Conf. on Information Processing in Sensor Networks (IPSN'03)*, 2003.

[19] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proc. of the 2nd ACM Intl. Conf. on Embedded Networked Sensor Systems (SenSys'04)*, Nov. 2004.

[20] R. Stoleru, T. He, and J. A. Stankovic. Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks. In *1st IEEE Workshop on Embedded Networked Sensors EmNetS-I*, October 2004.

[21] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler. An analysis of a large scale habitat monitoring application. In *Proc. of the 2nd ACM Intl. Conf. on Embedded Networked Sensor Systems (SenSys'04)*.

[22] Q. Wang, W. Chen, R. Zheng, K. Lee, and L. Sha. Acoustic target tracking using tiny wireless sensor devices. In *Proc. of 2nd Intl. Conf. on Information Processing in Sensor Networks (IPSN'03)*, 2003.

[23] N. Xu. Implementation of data compression and FFT in TinyOS. *http://enl.usc.edu/ ningxu/papers/lzfft.pdf*.

[24] P. Zhang, C. Sadler, S. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proc. of the 2nd ACM Intl. Conf. on Embedded Networked Sensor Systems (SenSys'04)*, Nov. 2004.

[25] F. Zhao, J. Liu, L. Guibas, and J. Reich. Collaborative signal and information processing: An information directed approach. *Proceedings of the IEEE*, 91(8):1199–1209, 2003.