12-2004

High Accuracy Distributed Target Detection and Classification in Sensor Networks Based on Mobile Agent Framework

Xiaoling Wang
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Xiaoling Wang entitled "High Accuracy Distributed Target Detection and Classification in Sensor Networks Based on Mobile Agent Framework." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Hairong Qi, Major Professor

We have read this dissertation and recommend its acceptance:

Mongi A. Abidi, Michael J. Roberts. Daniel B. Koch, Hamparsum Bozdogan

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Xiaoling Wang entitled “High Accuracy Distributed Target Detection and Classification in Sensor Networks Based on Mobile Agent Framework.” I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Hairong Qi
Major Professor

We have read this dissertation
and recommend its acceptance:

Mongi A. Abidi

Michael J. Roberts

Daniel B. Koch

Hamparsum Bozdogan

Accepted for the Council:

Anne Mayhew
Vice Chancellor and Dean of
Graduate Studies

(Original signatures are on file with official student records.)

High Accuracy Distributed Target Detection and Classification in Sensor Networks Based on Mobile Agent Framework

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Xiaoling Wang

December, 2004

Acknowledgments

I would like to extend my sincere gratitude and appreciation to all the individuals who made this dissertation possible.

First and foremost, I am grateful to my adviser Dr. Hairong Qi. Her guidance and encouragement through my studies has allowed me to develop my skills as a researcher. Without her support, this work would not have been possible. Also I am highly indebted to Dr. Hamparsum Bozdogan, Dr. Mongi Abidi, Dr. Michael Roberts, and Dr. Daniel Koch for serving as members of my committee and providing all the enlightening suggestions.

Special acknowledgment is given to Mr. Steve Beck, Mr. Joe Reynolds, and Ms. Carol Brewer at BAE Systems, Inc. for their elaborate work helping us set up the field demo at Austin, TX and collect the valuable experimental data. I also appreciate the work by Professor Yu Hen Hu and his students at University of Wisconsin-Madison for generating the cross-validation data set based on the SITEX02 field demo.

Last but not the least, I sincerely thank my husband Hongtao, my parents, my parents-in-law, my brother and sister for their unconditional support, love, and encouragement through my life.

Publication History

This dissertation appears in part in the following academic journals and conferences.

- **X. Wang**, H. Qi, S. Beck (2004). "Distributed multi-target detection in sensor networks", *Frontiers in Distributed Sensor Networks*. Editors: R. Brooks, S. S. Iyengar, CRC Press.
- **X. Wang**, H. Qi, S. Beck, H. Du (2004). "A progressive approach to distributed multiple target detection in sensor networks", *Sensor Network Operations*. Editor: S. Phoha, IEEE Press.
- **X. Wang**, H. Qi, Y. Tian, and S. S. Iyengar (2004). "Collaborative signal and information processing hierarchy in distributed sensor networks", *Submitted to International Journal on Distributed Sensor Network*.
- H. Qi, Y. Xu, and **X. Wang** (2003). "Mobile-agent-based collaborative signal and information processing in sensor networks", *Proceedings of IEEE*, 91(8): 1172-1183, August.
- H. Qi, **X. Wang**, S. S. Iyengar, and K. Chakrabarty (2002). "High performance sensor integration in distributed sensor networks using mobile agents," *International Journal of High Performance Computing Applications*, 16(3): 325-336, August.
- **X. Wang**, H. Qi (2004). "Collaborative unknown target recognition in sensor networks", *The Nineteenth National Conference on Artificial Intelligence, Workshop on Sensor Networks*, San Jose, CA, July.
- **X. Wang**, H. Qi (2004). "Mobile agent based progressive multiple target detection in sensor networks", *International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May.
- **X. Wang**, H. Qi, H. Du (2003). "Distributed source number estimation for multiple target detection in sensor networks", *IEEE Workshop on Statistical Signal Processing*,

St. Louis, MO, September.

- **X. Wang**, H. Qi, and S. S. Iyengar (2002). "Collaborative multi-modality target classification in distributed sensor networks," *International Conference on Information Fusion*, pp.285-290, Annapolis, MA, July.

- **X. Wang**, and H. Qi (2002). "Acoustic target classification using distributed sensor arrays,", *International Conference on Acoustics, Speech and Signal Processing*, Orlando, FL, May.

- Y. Tian, H. Qi, and **X. Wang** (2002). "Target detection and classification using seismic signal processing in unattended ground sensor systems," *International Conference on Acoustics, Speech and Signal Processing*, Orlando, FL, May.

- H. Qi, **X. Wang**, S. S. Iyengar, and K. Chakrabarty (2001). "Multisensor data fusion in distributed sensor networks using mobile agents," *Information Fusion*, TuC2-11-16, Canada, August.

Abstract

High-accuracy distributed information exploitation plays an important role in sensor networks. This dissertation describes a mobile-agent-based framework for target detection and classification in sensor networks. Specifically, we tackle the challenging problems of multiple-target detection, high-fidelity target classification, and unknown-target identification.

In this dissertation, we present a *progressive* multiple-target detection approach to estimate the number of targets sequentially and implement it using a mobile-agent framework. To further improve the performance, we present a *cluster-based distributed* approach where the estimated results from different clusters are fused. Experimental results show that the distributed scheme with the Bayesian fusion method have better performance in the sense that they have the highest detection probability and the most stable performance. In addition, the progressive intra-cluster estimation can reduce data transmission by 83.22% and conserve energy by 81.64% compared to the centralized scheme.

For collaborative target classification, we develop a general purpose *multi-modality, multi-sensor* fusion hierarchy for information integration in sensor networks. The hierarchy is composed of four levels of enabling algorithms: local signal processing, temporal fusion, multi-modality fusion, and multi-sensor fusion using a mobile-agent-based framework. The fusion hierarchy ensures fault tolerance and thus generates robust results. In the meanwhile, it also takes into account energy efficiency. Experimental results based on two field demos show constant improvement of classification accuracy over different levels of the hierarchy.

Unknown target identification in sensor networks corresponds to the capability of detecting targets without any *a priori* information, and of modifying the knowledge base dynamically. In this dissertation, we present a collaborative method to solve this problem among multiple sensors. When applied to the military vehicles data set collected in a field demo, about 80% unknown target samples can be recognized correctly, while the known target classification accuracy stays above 95%.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As the direct interconnection between human being and the physical environment, sensors and actuators link the world of events, tangible things, and organic creatures with the electronic world of computers, processors, and storage devices [4, 125, 141]. Recent advances in micro electrical mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that integrate sensing, processing, and communication capabilities together and form an autonomous entity. Large numbers of these sensors can be quickly deployed in the field, where each sensor node senses the environment independently, but as a group, the sensor nodes collaboratively achieve complex information gathering and dissemination tasks like area surveillance and environmental monitoring.

So far, researchers have drawn different pictures about the future of sensor networks. In an interesting article [99], K. Pister describes his vision of sensor networks in 2010, "In 2010 your house and office will be aware of your presence, and even orientation, in a given room. Lighting, heating, and other comforts will be adjusted accordingly. ... In 2010 a speck of dust on each of your fingernails will continuously transmit fingertip motion to your computer. Your computer will understand when you type, point, click, gesture, sculpt, or play air guitar ...".

Another view is provided by the founders of Ember Corporation (a startup company aiming at the market for "extremely low-cost, wireless 'thing-to-thing' networks for countless embedded processors, sensors, and controls" [125]), who envision a future when "every vine in a vineyard reports sunlight, temperature, and moisture every hour of the day, [while] every city street lamp monitors the passage of each bus and relays information ahead to waiting passengers". Even though the social implications of the move to integrate sensing and processing are enormous and varied, there is no doubt that sensor networks will become ubiquitous and change people's life dramatically.

This dissertation presents the details of our research in sensor networks and its applications in collaborative information processing. The remainder of this chapter first reviews the state-of-the-art of sensor networks in Sec. 1.1, including their potentials, challenges, application areas, and the available sensor networks developed for different application-specific tasks. In Sec. 1.2, the general architecture of a sensor node in sensor networks is presented. Similar to the design and implementation of computer networks, sensor network protocols are divided into different layers. A review of the sensor network protocol stack is given in Sec. 1.3, which discusses the protocols in the physical, data link, network, transport, and application layers with power management, mobility management, and task management functions. In Sec. 1.4, the computing paradigms of sensor networks in the application layer are discussed in further detail. Finally, corresponding to the focus of this dissertation, a review of collaborative signal and information processing (CSIP) algorithms is presented in Sec. 1.5. The contributions of this work are depicted, and the system components and dissertation organization are outlined at the end of the chapter.

1.1 Sensor Networks - State of the Art

A sensor network is composed of a large number of sensor nodes that are densely deployed either inside a phenomenon or very close to it. Each sensor node only has a limited amount

of sensing and processing capabilities. However, they function much like individual ants that, when they coordinate among themselves into a network, collaboratively accomplish complex tasks, enable real-time adaptation to environmental and user conditions, and provide capabilities greater than the sum of the individual parts [4, 45, 125].

To motivate the discussion of designing sensor networks, a typical scenario is considered. Thousands of sensor nodes are deployed randomly to cover an open field (for example, air-dropped from a helicopter). Once the sensors are in place, they coordinate to establish a communication network automatically, monitor the field of interest collaboratively in an energy-efficient manner, adjust their subsequent operations according to the remaining resources, and re-organize upon changes such as node failure, node addition, and node movement.

1.1.1 Potentials of Sensor Networks

The use of sensor networks has many potential advantages over a stand-alone single sensor deployment, which can be addressed from five aspects: redundancy, complementarity, localization, timeliness and cost [83].

Redundancy: When each sensor in a sensor network captures the same set of features of an environment with different fidelity, it provides redundant information about the environment. The fusion of information among multiple sensors will reduce the overall uncertainty and ensure greater signal to noise ratio (SNR), thus improving the accuracy. Furthermore, this high level of redundancy allows better fault tolerance of the system [19].

Complementarity: Complementary sensor networks may give a representation of data over a large spatial area through the union of many small sensor coverage areas, or provide several aspects of the same phenomenon that can be used together for studying one event which is otherwise impossible to perceive using individual sensors [20, 83]. It also helps sculpting the coverage for a given application and a given terrain to overcome shadows or holes. The usage of multiple sensor types can compensate the limitations of individual sensing modalities and

improve sensing performance.

Localization: Since sensor nodes are usually placed in close proximity to the object of interest, the high SNR obtained by each sensor node helps overcome ambient environmental effects. Furthermore, the short-range sensor technology provides the capability of sensing phenomena only at close distances and localizing discrete events through multiple sensor fusion [4].

Timeliness: By fusing the information from multiple sensors, it is possible to achieve higher processing speed due to the processing parallelism of the fusion algorithm [83].

Cost: The advances in sensor technology and MEMS fabrication have resulted in better, cheaper, and smaller sensors. This, on the other hand, has made it economically feasible to deploy large numbers of sensors in sensor networks.

1.1.2 Challenges of Sensor Networks

In spite of the potentials of sensor networks, it has presented unique challenges to many aspects of network design and information processing. The random-deployable characteristic of the sensor network makes it ad hoc, therefore, many protocols and algorithms developed for wireless ad hoc networks can be adopted into the design of sensor networks. However, there are several differences between sensor networks and ad hoc networks, which are summarized in [5, 112]:

- Sensor networks are composed of hundreds or thousands of sensor nodes that are designed for unattended operation.
- Sensor nodes in sensor networks are densely deployed and generally stationary after deployment except for some special mobile nodes.
- Sensor nodes are prone to failure. Therefore, the topology of a sensor network changes frequently due to node failure.

- Sensor nodes are limited in power since they are usually supplied on battery which cannot be replaced or recharged from remote operation. Therefore, the most important issue in sensor networks is to conserve energy and prolong the lifetime of the whole system.

The unique features of sensor networks raise several technical challenges that must be overcome before they can be used for today's increasingly complex information gathering tasks:

The fault tolerance challenge. Some sensor nodes may fail or be blocked due to lack of power, physical damage or environmental interference. However, the failure of sensor nodes should not significantly affect the performance of the whole sensor network. This is referred to as the *reliability* or *fault tolerance* challenge [5]. In the literature, fault tolerance is defined as the ability to sustain sensor network functions without any interruption due to sensor node failures [124].

The self-organization challenge. In some applications, sensor nodes are deployed randomly and rapidly, for example, from an aircraft, in remote terrain. This ad hoc deployment requires sensor nodes to communicate with their neighbors and set up the network automatically. In the case of node failures, sensor nodes in the network should be able to change the network topology adaptively and reorganize the available nodes to accomplish the task.

The scalability challenge. The performance of the sensor network should not be affected when the number of sensor nodes is increased.

The energy efficiency challenge. Wireless sensor nodes are normally supplied with batteries and equipped with limited power resource. In many application scenarios, it is impossible to recharge the battery. Therefore, developing energy efficient algorithms and protocols to prolong the lifetime of sensor nodes plays a critical role in sensor network applications. Moreover, in a multi-hop ad hoc sensor network, each node plays a dual role of data originator and data router. The malfunctioning of a few nodes can cause significant topological changes and thus require rerouting of packets and reorganization of the network [5]. From this point of view, energy conservation takes on additional importance.

In addition, several other primary requirements of sensor networks include: rugged packaging, easy deployment, self-localization, task adaptivity, and environment adaptivity. These challenges call for a re-consideration of the design of sensor networks, along with the underlying algorithms and protocols.

1.1.3 Applications of Sensor Networks

Research on sensor networks was originally motivated by military applications. However, the availability of low-cost sensors and communication networks has extended its application into a broad spectrum, from industrial to environmental and civilian monitoring.

Military Applications

Military application of sensor networks dates back to the time of the Cold War. Although the penalties for failure are high, the military has the organizational and disciplinary structure to deploy and utilize wireless sensors successfully [125].

The rapid deployment, self-organization and fault tolerance characteristics make sensor networks a promising technique for military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting (C4ISRT) systems [6]. Typical sensors useful for military scenarios range from air-launched, long-range acoustic/seismic sensors, to short-range, multiple-modality, networked sensors for placement by personnel or unmanned aerial vehicles (UAVs). Different sensors are connected through wireless links to monitor the status of a friendly force, equipment and ammunition, to obtain information on the composition and movements of opposing forces, to survey critical terrains, approach routes and paths, and to detect nuclear, biological and chemical weapons launched by enemies. Sensor networks need to fuse the data from multiple sources to ensure that users are not inundated with misinformation or too much information.

Industrial Applications

Commercial industry has long been interested in sensor deployment as a means of lowering cost and improving machine performance and maintainability. For several decades, sensors have been placed on or in machines to monitor their health through wear, heat, lubrication levels, or similar information [28]. Recently, with the advances in sensor technology, distributed computing and wireless communication technique, sensors begin to play an even more important role in manufacturing and monitoring effort. For example, as summarized in the "Buyer's Guide 2002" [125] (a trade publication in existence for 18 years), there are 79 technologies used in making and employing sensors, 116 physical properties that can be sensed, and more than 1900 suppliers, manufacturers, and solutions providers existed in the industry.

So far, factories have continued to automate production and assembly lines with remote sensing networks, thus increasing efficiency and decreasing resource waste. Meanwhile, the products can be instrumented to report machine and product status in real time for process control feedback and condition-based equipment maintenance. In the automobile industry, sensor networks are used to detect the pollution levels in an exhaust system when the engine runs, and to sense the vehicle deceleration for airbag deployment triggers [125]. Another important sensor network application in industry is structural health monitoring (SHM) where sensors are placed directly into the frame of buildings to detect structural damage and are connected wirelessly to exchange information [21]. The sensors used in SHM normally include strain gauges, accelerometers, and velocity sensors. These embedded sensors report on the structural integrity and strength of the building and send out alarm when the detected changes exceed its normal range.

Environmental and Habitat Monitoring

Environmental and habitat monitoring is a natural application of sensor networks since the variables to be measured are usually distributed in a large area. Environmental sensor networks can

be used to monitor the air [40], soil, and water [149]. In agriculture, wireless sensor networks are deployed to monitor both short-term changes in fertilizer and pesticide levels and long-term trend of vegetation responses to climate and diseases. Sensor nodes equipped with chemical sensors can be used to detect biological or chemical toxins in air or drinking water and provide early warning of attacks. A futuristic application is sensors worn like wristwatches to provide individual sensing of chemical or biological agents [125]. Sensor networks are also used in ecosystem monitoring which aims at understanding the response of wild populations to habitats over time [23, 145]. Other environmental sensor network applications include forest fire detection, flood detection, large-scale Earth monitoring and planetary exploration, and biocomplexity mapping of the environment [6].

Civilian Applications

The most significant application of sensor networks in civilian arena is telemedical care of mobile patients. Portable sensors can be carried by patients to provide continual reading on vital signs, blood pressure, and other physiological data and to determine medication levels in real time. In addition, sensor networks have been used to monitor transportation patterns and perform traffic control in urban areas [52, 60, 68]. The Intelligent Transportation Project conducted by Muntz et al. is one of the examples [46]. The smart Kindergarten project (Sensor-based Wireless Networks of Toys for Smart Developmental Problem-solving Environments) conducted by Srivastava et al. in UCLA [131] implements the sensor network technique, middle-ware services design and data management to realize the early childhood education environment investigation by monitoring the learning process through portable badges and networked toys which embed sensors connected to a control center through the wireless network. Sensor networks are also used in home automation, virtual keyboard implementation, environmental control in office building, etc [125].

In fact, due to the pervasive nature of sensors, sensor networks have the potential to revolu-

tionize the very way we understand and construct complex physical systems [45].

1.1.4 Sensor Networks in Action

Corresponding to the continuous improvement of sensor network technologies and the wide spread of its application in different aspects of human society, many ongoing research programs pursue these technologies and seek to extend their capabilities.

The Defense Advanced Research Projects Agency (DARPA) has been actively funding projects in sensor network research. The Distributed Sensor Networks (DSN) program started around 1980 is one of the first efforts which assumed many spatially-distributed low-cost sensing nodes collaborating with each other but operating autonomously, with information being routed to whichever node can best use the information [28]. For demonstration, two test beds were developed: a real-time test bed for acoustic tracking of low-flying aircraft at MIT Lincoln Laboratory and a distributed vehicle monitoring test bed at the University of Massachusetts, Amherst. More recently, DARPA sponsored the Sensor Information Technology (SensIT) program [72] which pursued two key research and development thrusts, energy-efficient networking techniques and networked information processing. Other similar programs include National Science Foundation (NSF)'s Sensors and Sensor Networks program starting from 2003 which seeks to advance fundamental knowledge in the areas of sensor design, materials and concepts, including sensors for toxic chemicals, explosives and biological agents, sensor networking systems in a distributed environment, the integration of sensors into engineered systems, and the interpretation and use of sensor data in decision-making processes [47].

The SmartDust project at UC-Berkeley pushes the size limit of sensors to an extreme - a cubic millimeter, such that these sensors can float in the air like dust [100]. The WINS (Wireless Integrated Network Sensors) project at UCLA [102] and the WSN (Wireless Sensing Network) project at Rockwell Science Center [116] integrate multi-modality sensing devices and low-level signal processor on the microsensor, making it more intelligent and powerful. Oceana

Sensor Technologies (OST) has developed an ICHM (Intelligent Component Health Monitor) system, which is a smart, networked, open-architecture sensor infrastructure, for prognostics and health management of aircraft engines and industrial machines [91]. The Piconet project aims at developing a prototype embedded network for personal location, equipment tracking, or information gathering [14]. The Ubiquitous Computing project at Xerox PARC explored a generalized version of these applications: seamless integration of computing devices into the environment [45, 142].

As time goes by, more and more projects are established to elaborate in-depth sensor network research and its applications in different areas. There exists by no means an exhaustive list of ongoing research on sensor networks. However, not many of them have been put out in the real world.

The Sound Surveillance System (SOSUS), a system of acoustic sensors on the ocean bottom, was deployed at strategic locations to detect and track quiet Soviet submarines during the Cold War. It is now used by the National Oceanographic and Atmospheric Administration (NPAA) for monitoring events in the ocean, e.g., seismic and animal activities [92]. Around the same time, networks of air defense radars were developed and deployed to monitor the continental United States and Canada. This air defense system has evolved over the years to include aerostats as sensors and Airborne Warning and Control System (AWACS) planes, and is also used for drug interdiction [28]. More recently, the government of Brazil sponsored a large sensor network, the System for the Vigilance of the Amazon (SIVAM), which interconnects different types of sensors (video, radar, and environmental), to provide environmental monitoring, drug trafficking monitoring, and air traffic control for the Amazon Basin [28]. Since 2002, researchers from Intel Research Laboratory at Berkeley have been working on a petrel project, where they employed hundreds of motes (a sensor platform built by Crossbow Technology Inc.) on Great Duck Island, Maine to monitor the life of a kind of small and secretive seabird known as the Leach's storm petrel [71]. After the motes are buried underground in the bird's burrow

or placed on the ground to measure weather conditions, they can self-organize into a wireless network and send their sensing data to a base station. Through satellite communication, the data can be accessed and further processed remotely from the Intel Lab at Berkeley.

The implementation of sensor networks in real world is still primitive. As Mainwaring points out, "You try to anticipate all the contingencies. Then you head out to the field, and some of them never happen, and others happen you hadn't even thought of" [71]. There are bumps along the way, to be sure. In this circumstances, more efforts have to be devoted in the design and development of realistic architectures of sensor networks.

1.2 Sensor Node Architecture

Sensor networks are usually composed of large number of sensor nodes connecting through wireless links. In general, a sensor node is a platform that combines sensing, data processing, wireless communication, and power components. It may also have additional application-dependent components such as a location finding system, power generator, mobilizer, etc. The sensing unit usually includes different kinds of sensing modalities and the analog-to-digital converters (ADC). The analog signals captured by the sensors are converted into digital data and then fed into the processing unit. The processing unit, which consists of a processor and an associated storage unit, processes data locally and collaborates with other sensor nodes to accomplish required tasks. The transceiver unit is used to communicate with other parts of the sensor network. The most important component of a sensor node is the power unit, which provides the energy resource to all the other components. Since there are some protocols and algorithms (for example, beamforming in target tracking) that require accurate information of the sensor node location, the location finding system is also commonly used to achieve location acquisition task. A mobilizer is useful if the sensor node needs to be moved in some applications in order to carry out the assigned task. The architecture of a sensor node is illustrated in Fig. 1.1 [5].

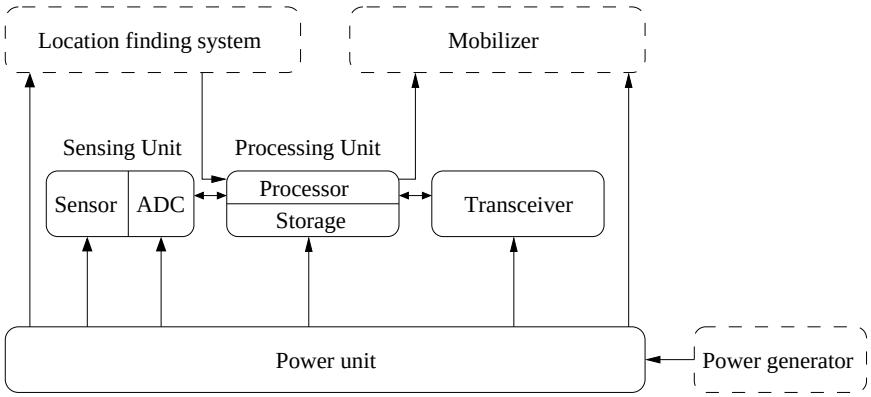


Figure 1.1: Sensor node architecture [5].

So far, a couple of commercial companies are building and deploying sensor platforms for various applications. Some examples of existing platforms include the mote-based testbed from Crossbow Corporation [35], the sGate sensor platform from Sensoria Corp [34], the SensorView System from Ricciardi Technologies, Inc. (RTI) [118], etc.

The motes include three generations of tiny, smart, wireless sensor platforms with sensing, processing, and communication capabilities. They are developed by UC Berkeley's research group on wireless sensors and commercialized by Crossbow Corporation (the second generation, MICA, and the third generation, MICA2 and MICA2DOT, are shown in Fig. 1.2). All of these platforms provide a plug-in sensor board with a processor running an event-driven TinyOS distributed software operating system and a two-way radio transceiver. For MICA, optical, thermal, seismic/acceleration, acoustic, and magnetic sensors can be equipped. While in the third generation, RH (relative humidity), barometric pressure sensing are added in. A very successful example of mote-based testbed is the Diuturnity testbed [139] developed at the University of Southern California.

The Sensoria sGate development platform employs a dual-issue Hitachi SH-4 processor as both a real-time interface and a power efficient RISC processor hosting 32-bit applications. In addition, a dual-mode RF modem system enables scalable wireless communication. On the

Figure 1.2: The Motes from Crossbow Corporation. Left: MICA; Middle: MICA2; Right: MICA2DOT [35].

sGate platform, the Linux operating system is used and up to four analog sensor inputs are supported. Common sensing modalities used on sGate platforms include acoustic, seismic, and PIR (passive Infra-red) sensing. This sensor platform is adopted in the DARPA SensIT program [12] to detect, track, and classify moving military targets. Figure 1.3 shows the Sensoria sGate platform.

The SensorView system (shown in Fig. 1.4) is a chemical-biological point detection system infrastructure, created by software development company, RTI [118]. It can be effectively employed in biological warfare and enables flexible control of disparate detectors, collectors, identifiers, and triggers to deliver early-warning detection, identification and communication of biological warfare agents (BWAs) and events [130]. The SensorView system is developed based on embedded PC/104 hardware components from Parvus Corporation where PC/104 is a standard for PC-compatible modules (circuit boards) that can be stacked together to create an embedded system. On top of PC/104, the system is able to embed PC architecture without having to use a bulky, less reliable motherboard- or backplane-based approach. The SensorView system integrates a variety of nuclear, biological and chemical sensors being deployed at various sensitive locations and performs the information acquisition task.

Figure 1.3: The Sensoria sGate platform [34].

Figure 1.4: The PC/104 based SensorView system [130].

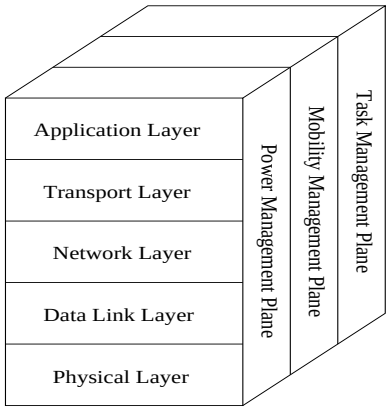


Figure 1.5: The protocol stack of sensor networks [5].

1.3 Protocol Stack of Sensor Networks

Similar to computer networks, sensor network design can be divided into different layers. Each layer accomplishes specific functionalities which are transparent to other layers. Different layers communicate through interfaces between each other. As defined in [5], the protocol stack of sensor networks consists of physical layer, data link layer, network layer, transport layer, and application layer, with power management, mobility management, and task management functionality. The protocol stack is illustrated in Fig. 1.5.

1.3.1 Physical Layer

The physical layer is a largely unexplored area in the design of sensor networks. It is responsible for power-aware modulation and hardware design instead of targeting high data rates as in most other communication systems.

Simple and low-power modulation schemes need to be developed for reliable communication in a sensor network. The modulation can be either baseband (any frequency band that is not shifted to some other frequency band but remains at its original place in the electromagnetic spectrum) or passband (a band of frequencies that is transmitted with maximum efficiency). In

[126], a comparison is conducted between binary and M-ary modulation schemes which shows that binary modulation techniques are more energy efficient under start-up dominant conditions encountered in low-power short range wireless transceivers. It is well known that long distance wireless communication consumes more energy and needs more complicated circuitry in noisy environments with signal fading. Therefore, multi-hop communication is recommended in sensor networks to achieve energy-efficient communication. Additionally, adaptive transmit power and dynamic voltage scaling are a couple of energy-efficient hardware strategies for sensor networks.

1.3.2 Data Link Layer

The data link layer, which focuses on the design of medium access control (MAC) protocols, is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. There are two important issues to be considered in the design of MAC protocols for sensor networks: energy efficiency which concerns the energy spent both in listening to the channel and in packet transmission and forwarding, and bandwidth allocation fairness of end to end data flow.

The available MAC layer protocols in wireless ad hoc networks can be divided into two categories: contention-based and reservation-based methods. The contention-based schemes are designed for minimum delay and maximum throughput during the network communication. Some typical examples of contention-based MAC protocols include IEEE 802.11 and PAMAS (Power Aware Multi-Access protocol with Signaling). However, since the contention-based protocols need continuous sensing of the access channel and the resources are wasted whenever a collision occurs, they are not suitable for sensor networks [112]. The reservation-based schemes of channel access attempt to set up connections between nodes first by detecting the neighboring radios and then assign collision-free channels to links by assigning channel slots

to individual users in a hierarchical manner through communication cluster formation and cluster heads assignment. Compared to contention-based protocols, reservation-based protocols have a lower energy consumption but more hardware requirement (eg. frame synchronization). TDMA (time division multiple access) mechanism is a natural choice of reservation-based protocols that can be used in sensor networks. Other MAC protocols that are more appropriate for sensor networks are discussed below:

Self-Organizing Medium Access Control for Sensor Network (SMACS): SMACS [129] is designed for network startup and link layer organization. It is a distributed infrastructure-building protocol that enables a collection of nodes to discover their neighbors and establish transmission/reception schedules without the need for any local or global master nodes. In order to simplify the formation process, the neighbor discovery and channel assignment phases are combined in the SMACS protocol, i.e., a channel (a pair of time intervals, similar to slots in TDMA schedule) is assigned to a link immediately after the link's existence is discovered. Therefore, by the time all nodes hear all their neighbors, they will have formed a connected network. Furthermore, since there is a potential for time collisions with slots assigned to adjacent links, each link is required to operate on a different frequency which is randomly chosen from a large pool of possible choices when the links are formed. This is done under the assumption that the available bandwidth is relatively large. SMACS protocol uses a scheduled mode of communication to enable energy savings for the node, which means a node with an established link knows when to turn on its transceiver ahead of time to communicate with another node, and it will also turn off when no communication is scheduled.

Eavesdrop-And-Register (EAR) algorithm: In [129], the authors also described another MAC layer protocol which is called the EAR algorithm. The EAR algorithm enables seamless interconnection of mobile nodes in the field of stationary wireless nodes, and represents the mobility management aspect of the protocol. It is desired that the connection be setup with as few message exchanges as possible, therefore, the mobile nodes assume full responsibility of the

connection setup process and decide when to break the connections. The mobile node will form a registry of neighbors in order to keep a constant record of neighboring activity. Making or breaking a connection is based on the status of connections, as well as the location and mobility information inferred from the entries in the registry. The EAR algorithm makes use of four primary invitation messages: broadcast invite (BI), mobile invite (MI), mobile response (MR), and mobile disconnect (MD) to control making and breaking connections. Acknowledgments are avoided by taking appropriate precautions, such as timeouts, to prevent lost messages. EAR is designed to be transparent to the existing stationary protocols, such as SMACS, which allows the functionality of the stationary protocol to remain fixed until the interjection of a mobile node.

1.3.3 Network Layer

Network layer design of sensor networks focuses on seeking an optimal routing protocol which can minimize a given metric, such as energy, number of hops, etc. According to [5], the network layer of sensor networks is usually designed following several principles: 1) Power efficiency is always an important consideration; 2) Sensor networks are mostly data-centric in that all communication is for named data, i.e., routing is based on data contained in the sensor nodes rather than traditional IP protocol where end-to-end delivery method is used based on unique identifications; 3) An ideal sensor network has attribute-based addressing and location awareness; 4) Data aggregation is useful only when it does not hinder the collaborative effort of the sensor nodes.

In general, routing protocols can be divided into flat routing and hierarchical routing protocols. Flat routing protocols can be further divided into multi-hop and cooperative network routing [112]. Multi-hop routing protocol deals with the routing between a source and a sink which is a node receiving data from other sensor nodes and might be communicating with the task manager node via Internet or satellite. Cooperative routing protocols involve cooperation

between sensors that detect a common target and send it to a central node for aggregation. So far, many routing protocols have been developed for sensor networks in favor of energy efficiency. Some popular examples supporting multi-hop communication scheme include:

Sequential Assignment Routing (SAR): The SAR protocol developed in [129] is a routing protocol that aims at minimizing the average weighted QoS (Quality of Service) metric throughout the lifetime of the network. In order to create multiple paths from each node to the sink, multiple trees that root from a one-hop neighbor of the sink are built. Each tree grows outward from the sink by successively branching to neighbors at higher hop distances from the sink while avoiding nodes with very low QoS and energy reserves. At the end of the process, most nodes will belong to multiple trees and thus have multiple paths disjoint inside the one-hop neighborhood of the sink. The advantage of SAR protocol is that it allows each sensor's indirect control of which one-hop neighbor of the sink will relay a message. There are two parameters associated with each path of the sensor nodes: estimated energy resource by maximum number of packets that can be routed, and additive QoS metric where a higher metric implies lower QoS. The SAR protocol takes into consideration the energy resource and QoS on each path, and the priority level of a packet for path selection for each sensor. Therefore, each sensor node that generates the packet selects a path to route the data back to the sink.

Sensor Protocols for Information via Negotiation (SPIN): Another popular routing protocol for wireless sensor networks is the SPIN developed at MIT [57]. SPIN is data-centric in that data are named using high-level data descriptors, the meta-data. The meta-data and the raw data have a one-to-one mapping relation. Considering that classic flooding suffers from the problem of implosion in that information is sent to all nodes regardless of whether they have already seen that information or not, SPIN addresses the deficiencies of classic flooding by using meta-data negotiation to eliminate redundant data transmissions over the network. SPIN is designed based on two basic ideas: sensor nodes operate more efficiently and conserve energy by sending meta-data instead of sending the raw data. There are three types of messages used

by SPIN, ADV, REQ, and DATA. The initiating node with new data advertises the data to its neighbors in the network by sending an ADV message containing a descriptor (meta-data). If the neighbor node needs the data, it sends a REQ message to the initiator for the advertised data and the initiator will send DATA message back to the neighbor. Therefore, SPIN is a sender-initiated routing protocol and all sensor nodes in the entire network that are interested in the data will get a copy.

Directed Diffusion: Directed diffusion [45, 64] developed at USC/ISI and UCLA is a scalable and robust network protocol for distributed sensor coordination. Directed diffusion is also data-centric and the data generated by sensor nodes are named using attribute-value pairs. A sensing task is disseminated throughout the sensor network from nodes that request data as an interest. If the attributes of the generated data match the interest, gradients will be set up within the network and data will be pulled toward the originator of interest along multiple paths, which forms a receiver-initiated routing protocol. An important feature of directed diffusion is that interest and data propagation and aggregation are determined by localized interactions (message exchanges between neighbors or nodes within some vicinity) [64]. Directed diffusion also facilitates the design of energy-efficient distributed sensing applications by providing Geographic and Energy Aware Routing (GEAR) protocol.

Besides multi-hop routing protocols, the flat routing protocol category also includes cooperative routing techniques: noncoherent and coherent routing. According to [129], for noncoherent routing, raw sensor data will be preprocessed at each node to extract a small set of parameters to be forwarded to a central node. In the contrast, for coherent routing, after minimal preprocessing, raw sensor data will be tagged with a timestamp and forwarded to the central node. Generally speaking, noncoherent techniques have lower data traffic and are more appropriate for sensor network implementations. However, coherent techniques still can be used with some path optimization techniques to achieve energy efficiency.

Different from the flat routing protocols, the hierarchical routing protocols are cluster hi-

erarchy based routing algorithms. One of the representative hierarchical routing protocols is LEACH proposed in [56].

Low-Energy Adaptive Clustering Hierarchy (LEACH): LEACH [56] is designed for sensor networks where an end-user wants to remotely monitor the environment. In such a situation, the data from individual nodes must be sent to a central base station which is often located far from the sensor network, through which the end-user can access the data. LEACH is a cluster-based protocol that includes distributed cluster formation, local processing to reduce global communication, and randomized rotation of the cluster heads to evenly distribute the energy load among sensors. The operation of LEACH is divided into two phases: setup phase and steady phase. During the setup phase, each node generates a random number and compares it with a pre-determined threshold to decide if it is a cluster head. After the cluster heads are selected, they announce their decisions to their neighbors. Once receiving the announcements, the neighboring sensors will determine which cluster to join based on the signal strengths of the received announcements and send back their acknowledgment to the corresponding cluster heads. Whereas, during the steady phase, sensor nodes perform sensing and transmit data to the corresponding cluster head. The cluster heads also aggregate data from their neighbors within the same cluster and send the fusion result to the base station. It is claimed that LEACH is energy efficient and it can extend the life of wireless sensor networks by a factor of 8 when compared to other multi-hop routing protocols.

1.3.4 Transport Layer

The development of transport layer protocols is a challenging problem, and so far few research efforts have been dedicated to this issue. However, this layer is especially needed when the system is planned to be accessed through the Internet or other external networks [5]. The traditional transport layer protocols for computer networks may be used, for example, TCP can be used to connect the sink nodes to the other networks such as the Internet or via satellite.

Communication between the sink nodes and sensor nodes can be carried through a UDP-type protocol. However, since many of the routing protocols in sensor networks are data-centric, i.e., using attribute-based naming instead of global addressing, UDP/TCP protocols cannot match the characteristics of sensor networks. Therefore, novel transport layer protocols are needed.

1.3.5 Application Layer

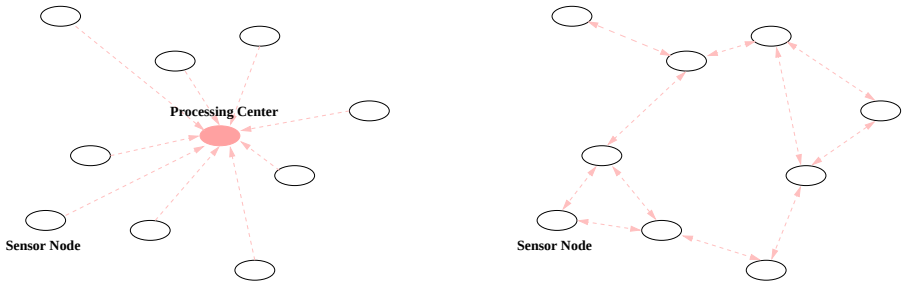
Although many application areas for sensor networks have been defined and proposed, potential application layer protocols for sensor networks remain a largely unexplored region till recently [5]. According to the survey conducted by Akyildiz et al., there are three possible application layer protocols developed so far: Sensor Management Protocol (SMP), Task Assignment and Data Advertisement Protocol (TADAP), and Sensor Query and Data Dissemination Protocol (SQDDP). SMP is used to connect sensor networks and the system administrator. Due to the unique characteristics of sensor networks, SMP needs to access the sensor nodes by using attribute-based naming and location-based addressing. TADAP is responsible for interest dissemination and data advertisement in sensor networks since users need to send their interests into the sensor network for a particular attribute and the sensors with data need to advertise their available data to the neighbors and users. SQDDP provides user applications with interfaces to issue queries, respond to queries and collect incoming replies.

Since distributed sensor networks are characterized by limited battery power, frequent node failure, and variable data and communication quality, these systems have to depend primarily on collaboration among distributed sensors in order to significantly improve sensing accuracy, reduce detection latency and scale up to more realistic battlefield tracking and classification applications involving multiple targets, heterogeneous sensing modalities, and non-uniform spatio-temporal scales. Collaborative signal and information processing (CSIP) is developed attempting to solve these problems. CSIP in distributed sensor networks is an emerging interdisciplinary research, drawing upon traditionally distinct disciplines such as lower-power

communication and computation, space-time signal processing, distributed and fault-tolerant algorithms, adaptive systems, sensor fusion and decision theory [73]. The objective of CSIP is to develop algorithms and approaches for representing, storing, and processing distributed, multi-modal information. A review of CSIP researches is presented in Sec. 1.5. The focus of this dissertation is on the discussion of CSIP in sensor networks featuring specific applications in target detection and classification.

1.3.6 Power Management, Mobility Management, and Task Management

The power, mobility and task management planes in the sensor network protocol stack monitor the power, movement, and task distribution among the sensor nodes which help the sensor nodes coordinate sensing tasks and reduce overall power consumption. As concluded in [5], the power management plane manages how power is used on each sensor node. For example, a sensor node may turn off its receiver after receiving a message from one of its neighbors to conserve energy. Or when the power level of a sensor node is low, it can broadcast to its neighbors that it is low in power and cannot participate in routing messages so that it can reserve the remaining power for sensing only [5]. The mobility management plane detects and registers the movement of sensor nodes in order to maintain a route to the user at all time and keeps track of neighboring relations among sensor nodes. With the knowledge of who the neighbor nodes are, sensor nodes can balance their power and task usage. The task management plane takes care of scheduling and balancing the sensing tasks given to a specific region since not all sensor nodes need to be activated at the same time. Therefore, depending on their available power level, some sensor nodes may be more active in performing tasks than others. With the functionalities provided by these management planes, the sensor nodes are able to work together in a power-efficient way, route data in a mobile environment, and share resources among sensor nodes to achieve complex information-processing tasks.



(a) Centralized client/server model.

(b) Decentralized peer-to-peer model.

Figure 1.6: The architectures of client/server and peer-to-peer models.

1.4 Computing Paradigms of Sensor Networks

In the context of sensor networks, computing paradigm refers to the information processing model deployed in the application layer of the protocol stack. No matter how different the sensor network topologies are, the computing paradigms of sensor networks can be mainly divided into two categories: centralized client/server model and decentralized peer-to-peer model. The architectures of both models are illustrated in Fig. 1.6.

1.4.1 Centralized Client/Server Model

The client/server model is one of the primary ideas of network computing. It is defined as a computational architecture that involves client processes requesting services from server processes. The client/server model provides a convenient way to interconnect programs that are distributed across different locations. Normally, the client's responsibility is to handle user interface, translate user's requests, send requests to the server, wait for server's response, translate the response into human-interpretable format, and finally present the result to the user. On the other hand, the server's functions consist of listening to requests, processing the received requests, and returning responses to the client. Many distributed systems have been developed

based on the client/server model, such as remote procedure calling (RPC) [15], common object request broker architecture (CORBA) [10], etc.

In sensor networks, the centralized client/server model is referred to as the process in which the clients (individual sensor nodes) send data to the server (processing center) where data processing tasks are carried out. Even though most sensor network protocols and data fusion algorithms are developed on top of the client/server model, it has several drawbacks that hinder its further applications. As discussed in [134], the drawbacks of client/server model in sensor networks can be concluded as follows: First, the client/server model may require many round trips to complete a transaction or query between a client and the server. Each trip creates network traffic and consumes bandwidth. In a system with many clients and/or many transactions, the total bandwidth requirements may exceed available bandwidth, resulting in poor performance for the application as a whole. Second, the design of a traditional client/server architecture requires decisions about where a particular piece of functionality will reside based on network bandwidth constraints, network traffic, transaction volume, number of clients and servers, and many other factors. If the estimates are not accurate, the performance of the application will suffer. Unfortunately, once the system is built and performance is measured, it is often difficult if not impossible to change the design. Third, the client/server model also requires the network connection to be alive and healthy the entire time a transaction or query is taking place. If the connection goes down, the client has to start the transaction from the beginning, if it can restart it at all. Considering the deficiencies of traditional centralized client/server model, new computing paradigms have to be developed to compensate these drawbacks.

1.4.2 Decentralized Peer-to-Peer Model

In large-scale distributed systems, the standard centralized client/server model presents two major problems: 1) Individual resources are concentrated on one or a small number of nodes. Therefore, sophisticated load balancing and fault-tolerance algorithms have to be applied in

order to provide real-time responses. 2) The transmission of data between clients and server consumes much energy. To solve these problems, decentralized peer-to-peer model is developed for distributing processing load and network bandwidth among all nodes participating in a distributed information system [2].

Peer-to-peer computing is the sharing of computer resources and services by direct exchange. More specifically, peer-to-peer is a communication model in which each party has the same capability and either party can initiate a communication session. In some cases, peer-to-peer communication is implemented by giving each node both server and client capabilities. As Bob Knighten, peer-to-peer evangelist of Intel states [33], "Peer-to-peer computing is the natural self-organizing complement to the centralized organization of client-server computing." Peer-to-peer is specifically effective for situations which require rich collaboration or disconnected use of digital assets. It is characterized by decentralized control, large scale, and extreme dynamics of its operating environment [95].

The peer-to-peer model has been the most popular computing paradigm in today's networking domain where the Internet may be the largest and the most successful example. More recently, the research in mobile ad hoc sensor networks uses the peer-to-peer paradigm. Conceptually, peer-to-peer systems exhibit several underlying principles, addressing the aspects of resource sharing, decentralization, and self-organization [2].

- The principle of sharing resources: All peer-to-peer systems need to share resources among their distributed components. The resources can be physical entities such as disk space or network bandwidth, as well as logical resources such as distributed services and different forms of knowledge. Through resource sharing, the network is able to achieve tasks beyond the capability of an individual node.

- The principle of decentralization: The immediate consequence of resource sharing in peer-to-peer systems is decentralization which means all components should present equal capabilities and the processes should no longer be conducted centrally. The decentraliza-

tion character can help the system to tolerate single-point failure and to avoid a transmission bottleneck.

- The principle of self-organization: Since peer-to-peer systems employ a decentralized structure, there is no central processing unit that can coordinate all the node activities and obtain the global information of the whole system. In this sense, the nodes need to self-organize themselves by interacting with their neighbors to exchange local information.

1.4.3 Mobile-Agent-Based Computing Paradigm

Agents are regarded as one of the possible solutions to the problem of effective parallel processing for large distributed systems. The agent paradigm is considered particularly good in coping with multiple requests in a peer-to-peer system since there is no precise priority or communication scheme that can be used to make the I/O flow serial [1]. In the context of sensor network applications, mobile-agent-based computing paradigm can be adopted to provide an efficient solution where data stay at local sites, while the integration code is transferred to data sites.

Generally speaking, a mobile agent is defined as a special kind of software that can execute autonomously. It moves to a host that contains data with which the agent wants to interact and takes advantage of being in the same host as the data. As discussed in [78], there are several good reasons to use mobile agent, such as reducing the network load, overcoming network latency, executing asynchronously and autonomously, etc. When implemented in sensor network applications, mobile-agent-based paradigm exhibits benefits as follows:

- Bandwidth efficiency. In sensor networks, sensor nodes are normally connected through wireless connections which provides much narrower bandwidth than wired connections. By using the mobile agent paradigm, instead of transmitting a large amount of raw data over the network through several round trips, only the agent (code) with small size is sent. This makes the efficient usage of network bandwidth possible.

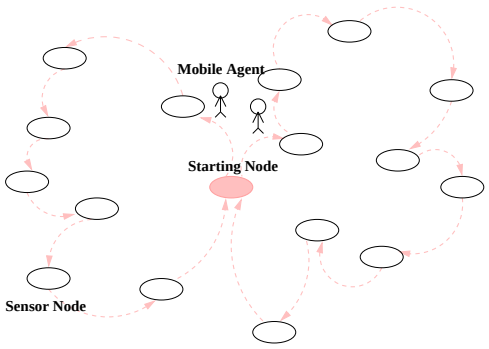


Figure 1.7: The architecture of mobile-agent-based paradigm.

- Better network scalability. Since mobile agents are able to adapt dynamically to distribute themselves among the nodes in the network in such a way as to maintain the optimal configuration for accomplishing a particular task [78], the performance of the network will not be affected when the number of sensors is changed.
- Stability. The execution of mobile agents can be asynchronous which means they can be sent when the network connection is alive and return results when the connection is re-established. Furthermore, the ability of mobile agents to react dynamically to unfavorable situations and events makes the whole network robust and fault-tolerant.
- Extensibility. Mobile agents can be programmed to carry task-adaptive processes which extends the capability of the system.

The architecture of mobile-agent-based paradigm is illustrated in Fig. 1.7.

1.5 Collaborative Signal and Information Processing in Sensor Networks

Collaborative signal and information processing (CSIP) in sensor networks has drawn more and more attention in recent years as an interdisciplinary research area which involves the study of

lower-power communication and computation, space-time signal processing, distributed algorithms, adaptive systems, and sensor fusion and decision theory [73]. The main concern of CSIP is to develop situational awareness using low-level sensor processing and local exchange of data to reach consensus in the neighborhood about the occurring events. It needs to be scalable, adaptive, energy-aware, and capable of delivering reliable information in real time. CSIP has been added as a technical area in many popular conferences and other special issues. For example, a workshop on collaborative signal processing sponsored by DARPA is held each year since 2001, the IEEE Signal Processing Magazine organized a special issue on CSIP in microsensor networks in March, 2002 which provides a snapshot of the state of the art in CSIP research for sensor networks, and the EURASIP Journal on Applied Signal Processing published a special issue in March, 2003 which is also focusing on CSIP in sensor networks.

The subject of CSIP ranges from detection, classification, and tracking of targets to energy efficiency, distributed compression, active sensor querying, high-level information processing, etc. CSIP in sensor networks provides a new paradigm to the signal and information processing community in contrast with single-platform signal processing. In the mean time, it presents some new challenges considering the constraints of application requirements, such as energy efficiency, network latency, and fault tolerance issues. According to [73], the primary research challenges in CSIP includes:

- Dense deployment of sensor nodes. Since thousands of sensor nodes are usually densely deployed in the field, it is possible for the sensor network to provide dense spatial sampling in multi-modality of phenomena of interest. Therefore, the challenge would be to combine the distributed data, first at each node and then with collaboration among the relevant devices in the network to produce meaningful global results.

- Asynchronous property. The distributed processing in a wireless sensor network typically is asynchronous, for example, in a sequential fusion center, the data from other sensor nodes may arrive out of order. This makes it necessary to design relevant signal and

information processing and fusion algorithms in order to deal with the asynchronous executions.

- Requirement of progressive accuracy. The sensor nodes in a sensor network are normally battery powered. The limited energy resource makes it important to develop power-aware signal processing and communication methods to provide progressive accuracy, such that the collaboration process could be terminated anytime upon achieving desired accuracy to conserve energy.

- Energy efficiency. The sensor network must optimize the trade-off between fault tolerance and energy efficiency in signal processing, data fusion, querying, and routing tasks in order to meet the energy constraints and at the same time to achieve good performance.

Being aware of the challenges posed on CSIP in sensor networks, more and more research groups have been actively working on different issues to make CSIP a practical and effective method for intelligence, surveillance and monitoring applications.

Energy-aware issues: Sensor nodes are normally battery powered and hence operate on an extremely limited energy resource. Further, they must have a lifetime on the order of months to years, since battery replacement is almost impossible for networks with thousands of physically embedded nodes [111]. Therefore, one of the most critical issues in sensor network is how to conserve energy and hence increase the lifetime of the entire sensor networks. Considerable research efforts have been devoted in incorporating energy awareness into every stage of sensor network design and operation. Specifically, the collaborative signal processing nature of sensor networks offers significant opportunities for energy conservation. [111] describes architectural and algorithmic approaches that perform aggressive energy optimization targeting all stages of sensor network design, from individual nodes, to wireless communications, and network-wide optimization. At the node level, with the sensor node architecture illustrated earlier in Fig. 1.1, the energy-aware techniques include 1) power-aware computing, for example, idle component

shutdown scheme in which the sensor node is shut down or sent into one of several low-power states if no events of interest occur, or dynamic voltage scaling (DVS) to dynamically adapt the processor's supply voltage and operating frequency according to the instantaneous computational load [96]; 2) energy-aware software, including the operating system, application layer, and network protocols; 3) power management of radios whose importance lies in the fact that wireless communication is a major power consumer during system operation; 4) energy-aware packet forwarding which reflects the functionality of sensor nodes as routers forwarding packets for other nodes. At the network-wide level energy optimization, one of the most important aspect is to conduct the communication between nodes in an energy efficient manner. From a wireless communication point of view, this can be achieved through an appropriate choice of modulation scheme such as quadrature amplitude modulation (QAM) and modulation scaling with respect to the traffic load. Furthermore, traffic distribution uniformization plays an important role in energy efficiency at the network level in the sense that it is preferable to avoid continuously forwarding traffic via the same path even though it minimizes the energy. Since sensor networks are densely deployed, an inherent redundancy exists among the data captured by multiple sensors. Combining the information from sensors detecting the same event of interest can both improve the reliability of detection and greatly reduce the amount of traffic, which in turn reduces the energy consumption on information communication.

Array signal processing: Kung Yao et al. from UCLA focus on implementing space-time processing (such as the blind beamforming algorithm) for target localization in sensor networks with random sensor deployment [26, 150]. The developed approaches are applied to deal with acoustic or seismic (i.e., vibrational) sources which are normally of wideband waveforms. Blind beamforming is an operation similar to conventional beamforming except without the knowledge of sensor responses and locations. Yao and his colleagues applied several sub-optimal space-time processing algorithms on source localization and DOA (direction of arrival) estimation, such as the closed-form least squares source localization, the classical MUSIC (multiple

signal classification) algorithm, and the maximum-likelihood parametric method. They also conducted several simulation scenarios and experiments to compare the performance of different algorithms. It is claimed that by using the maximum-likelihood algorithm developed, they can effectively locate multiple sources with random sensor placement.

Collaborative tracking, classification, and sensor fusion: Feng Zhao et al. from PARC (Palo Alto Research Center) developed an information-driven approach to dynamic sensor collaboration in ad hoc sensor networks. The main idea of the approach is for a network to determine participants in a sensor collaboration by dynamically optimizing the information utility of data for a given cost of communication and computation [151]. The proposed information-driven approach is applied to a task of tracking a moving vehicle through a two-dimensional sensor field. It is assumed that there is only one leader node active at any time and its task is to select and route tracking information to the next leader. In the context of target tracking, information utility is practically estimated as the usefulness of a measurement based only on characteristics of a sensor such as its location or sensing modality, which is estimated as the expected posterior distribution. By using measures of information utility, the sensors in a network can exploit the information content of data already received to optimize the utility of future sensing actions, and efficiently manage the scarce communication and processing resources. The problem of distributed tracking is formulated as a sequential Bayesian estimation problem and an information-driven sensor querying (IDSQ) framework is developed that tries to select a sensor which is likely to provide greatest improvement to the estimation of target state at the lowest cost.

Researchers from the University of Wisconsin-Madison developed a framework for distributed classification and tracking in sensor networks [38, 80]. The developed detection and tracking framework is based on the coordination between networking/routing protocols and collaborative signal processing algorithms. One of the key premises behind the networking algorithms developed at Wisconsin is that routing of information in a sensor network is geo-

graphic centric rather than node centric, which means compared to the arbitrary identities of sensor nodes, their geographic locations are the critical quantities. By dividing the geographic region of interest into smaller spatial cells, the nodes belonging to different cells can be activated at different times. Suppose some of the nodes in each cell are designated as manager nodes, at each time instant, the manager nodes in the activated cell determine the location of the target from the energy detector outputs of the active nodes and use the locations of the target at the previous successive time instants to predict its location at a specific future time instant. The predicted positions of the target are used to create new cells that the target is likely to enter. Once the target is detected in one of the new cells, it is designated as the new active cell and performs detection task consequently. This framework can be effectively used in single target tracking and the case where multiple targets are sufficiently separated in spatial-time field. In the presence of multiple targets that are close to each other, classification algorithms need to be performed. The problem of target classification is studied in the context of acoustic or seismic signals. Using features extracted from the spectra of signals (such as nonparametric or parametric power spectral density), different classifiers are implemented and their performances are compared. These classifiers include kNN (k-nearest-neighbor) classifier, maximum likelihood classifier using Gaussian mixture density model, and SVM (support vector machine) classifier. It is claimed that when classifying between tracked and wheeled vehicle, the best performance of the developed classifier can achieve 96%. Futhermore, on the collaboration methods between sensor nodes in target classification, two schemes, data-fusion and decision-fusion, are compared theoretically and it is shown that the decision-fusion method is more energy-efficient and suitable for sensor network applications.

Brooks, Phoha, and Friedlander from Penn State University developed the concept of reactive sensor network and semantic information fusion scheme for collaborative target tracking and classification [18, 20, 97]. The basic objective of the reactive sensor network framework is to support sensor network data aggregation and flexible tasking by applying mobile code

technologies. Mobile code, also known as downloadable code, is a software that is transmitted across the sensor network from a remote site to a local system and then executed on that local system without explicit action on the part of the user [31, 49]. For example, in the case of target classification in sensor networks, there may be several different classifiers available at a central site that are designed in favor of different target types. According to the initial decision of the target type from a default classifier, a corresponding classifier, which outperforms others for the given target type, can be downloaded to the local sensor node and replace the default classifier to perform classification locally. Semantic information fusion scheme deploys a hierarchical processing of sensor data. The raw time series data are processed within each sensor platform by identifying events from peaks in the raw data. Then the platforms in a geographical region self-organize into local neighborhoods and each platform transmits the local peaks to its neighbors. After comparing local peaks to the neighboring peaks within a space-time window on each platform, the one whose local peak is larger than its neighbors is chosen to process the event. Finally, the processed semantic data are transmitted across the network outside the neighborhood. This scheme works well for sensor network applications since 1) the abstraction of data into higher level semantics for information fusion and compression can reduce the bandwidth requirement; 2) the processing organized only around the geometry of target events conserves power.

Generally speaking, sensor networks are designed to be task-specific, i.e., to sense the environmental situations of interest and answer specific user queries of the form:"How many pedestrians do you observe in the observation field and in which geographical regions are they moving?", or "Which enemy vehicle is likely to be able to get close to building A first?" Such queries may also be a tool for organizing the sensor network itself and be able to activate sensors within a specified region. Therefore, the focus will be on tracking spatial or temporal relations between objects and local or global attributes of the environment, as opposed to the detailed estimation of positions and poses of individual objects [53]. It is claimed in [53] that high-level

behaviors of objects may be more robustly trackable than their exact positions, relations between objects may be easier to track than each object individually, and the large-scale behavior of an ensemble of objects may be easier to ascertain than the motion of the individual objects. The relations between objects in sensor network applications consist of leader-in-the-corridor which seeks to find the leader of a group of objects under given conditions, am-I-surrounded which is to decide if the object of interest is surrounded, i.e., if there is a line in the plane to separate the object from all others (say, enemy vehicles), target-counting to determine the number of objects within the sensing field which can be local or global, and cluster-maintenance in the mobile sensor devices scenario. Guibas considered the problem of computing answers to queries as a standard algorithm and proposed a mathematical theory of algorithm design that includes the cost of accessing the manifest variables of the problem or of determining useful atomic relationships among them [53]. It is argued that the advantage of the relation-based attribute computation is that the defining objects can move, but as long as the relations of interest among them stay valid, the attribute computation also remains valid. Therefore, it achieves energy conservation in sensing, processing, and communication in sensor networks.

Distributed signal processing: Many sensor network scenarios need to deploy sensor nodes densely to enable robustness to node failures. This induces a high level of redundancy between sensor data which are highly correlated. Several researchers are working on coding methods to reduce the sensor data redundancy so that the requirement of bandwidth and power consumed on information transmission could be minimized. Pradhan et al. proposed a constructive framework, called DISCUS (distributed source coding using syndromes), to remove the redundancy in a completely distributed manner, i.e., without the sensors talking to one another [103]. The DISCUS framework is designed based on fundamental concepts from information theory, which includes source coding, channel coding, and estimation theory. The distributed compression is considered as the problem of compressing an information source in the presence of side information only at the decoder in the form of another correlated source. The goal is

then to reconstruct the source at the decoder using the side information as well as the data sent by the encoder. According to [103, 104], the design of DISCUS includes: 1) source quantization and estimation for the desired distortion performance; 2) the representation codebook to maximize the correlation between the active codeword and the side information; 3) the channel code to have a large achievable channel rate with minimum probability of decoding error on the space of the source codebook; 4) efficient rule for decoding the side information in a given subset of the channel code. Through examples of sensor network information dissemination, it is shown that the DISCUS concept can reduce the cost of data transmission depending on the network topology and correlation structure, leading to the promise of significant network energy savings. It also provides a solution for the problem of optimal sensor fusion under bandwidth constraints.

1.6 Contributions

In this dissertation, we focus on the development of a number of agent-based distributed processing algorithms for different aspects of sensor network applications, multiple target detection and target recognition specifically.

Progressive multiple target detection. When considering different targets in the field as sources and assuming the signals they generate to be independent, multiple target detection in sensor networks can be solved using traditional blind source separation (BSS) algorithms where the signal captured by individual sensor is a linear/nonlinear weighted mixture of the signals generated by the sources. Due to the sheer amount of sensor nodes deployed in a sensor network, the fundamental assumption that the number of sources equals the number of sensors does not hold. Therefore, the problem of multiple target detection is converted into a source number estimation problem. Even though several techniques have been proposed for source number estimation, their usage in sensor networks is hindered by their centralized structure where the raw data from a large amount of sensors have to be transmitted to a central processing

unit. This process will generate significant data traffic over the network, occupy communication bandwidth, and consume energy. To solve this problem, we proposed a progressive detection approach based on the classic Bayesian estimation algorithm and implemented it using an agent-based framework. In contrast to the centralized scheme, the progressive approach sequentially estimates the number of targets based on only the local observation and partial estimation result from the previous sensors. By using agent-based framework, the processing code is carried by the mobile agent and executed autonomously at each stop. Furthermore, the mobile agent can dynamically decide its itinerary adapting to the changing environment. Experimental results on a civilian vehicle data set collected in a field demo (held at BAE Systems, Austin, TX) show that the distributed multiple target detection scheme using agent-based progressive approach has comparable estimation capability as that using centralized algorithm, but it reduces the amount of data transmission by 83.22% and conserves 81.64% of energy.

High accuracy collaborative target classification. For target classification, we designed a general purpose information processing and fusion hierarchy and implemented it in an agent-based framework. The hierarchy is composed of four levels of enabling algorithms: local signal processing, temporal fusion, multi-modality fusion of sensors on the same sensor node, and multi-sensor fusion across a cluster of sensor nodes. The fusion hierarchy ensures fault tolerance to handle uncertainty and hardware component failure, thus it can generate robust classification results. In the meanwhile, the agent-based implementation also takes into account energy efficiency which is the most contingent resource in sensor networks. Experimental results on DARPA SITEX02 data set show a steady increase in the classification accuracy across the different levels of fusion and the agent-based multi-sensor fusion always provides the highest accuracy.

Unknown target detection and classification. In the context of target recognition in sensor networks, the requirement of adding intelligence largely corresponds to the capability of recognizing unknown targets, i.e., targets without any a priori information, and of modifying

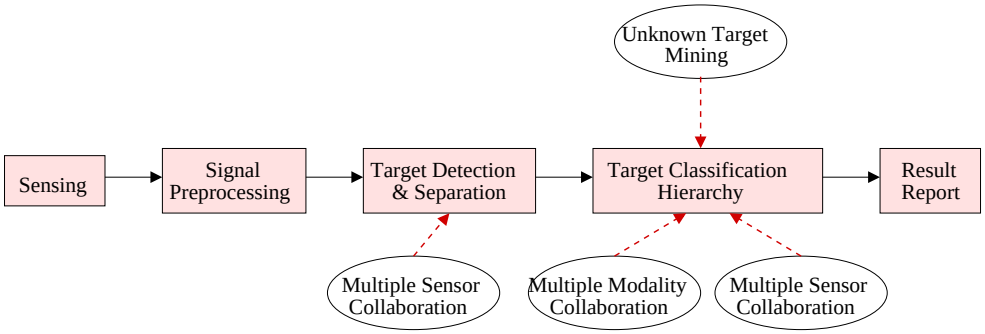


Figure 1.8: System modules.

the knowledge base dynamically to incorporate the newly discovered knowledge. To realize this functionality, we developed a collaborative unknown target recognition algorithm based on entropy estimation and data mining techniques. The collaboration among multiple sensor nodes is carried out using Dempster's fusion rule and can be implemented in an agent-based paradigm.

1.7 Document Organization

A complete CSIP system for target detection and classification includes the following modules: sensing, low-level signal preprocessing, detection and separation, target classification hierarchy with collaborative fusion, and real-time unknown pattern mining. The structure of the system is illustrated in Fig. 1.8.

The dissertation is organized based on the system structure:

Chapter 2 describes the physical mechanisms of several sensing modalities, acoustic, seismic/acceleration, infra-red, optical, and magnetic sensing. The focus of this work is on the employment of acoustic and seismic signals in collaborative target detection and classification. Several aspects of the acoustic and seismic sensing methodology are discussed, including the devices, the signal characteristics, etc. A review is also presented on acoustic/seismic signal processing approaches.

Chapter 3 focuses on the algorithm for multiple-target detection in sensor networks. A distributed source number estimation framework is developed, which includes two levels of processing: the local estimation is generated within each cluster using the mobile-agent-based progressive source number estimation method and a fusion algorithm is performed to combine the local results. A posterior probability fusion method is developed based on Bayes theorem and is compared with Dempster's rule of combination.

Chapter 4 presents a multi-modality, multi-sensor fusion hierarchy for target classification and information integration in distributed sensor networks. Each sensor node can individually sense its surroundings, but collaboratively achieve more complex information dissemination tasks. The hierarchy is composed of four levels of enabling algorithms: local signal processing, temporal fusion of local processing results over one sensing modality, multi-modality fusion at each sensor node, and multi-sensor fusion across a cluster of sensor nodes using a mobile-agent-based computing paradigm.

Chapter 5 discusses a collaborative unknown target recognition algorithm, which involves four enabling components: training set preprocessing, local node classification, collaborative classification based on local results derived from multiple sensors, and dynamic update of the training set when new target classes are discovered.

Chapter 6 exhibits experimental results based on the theoretical analysis in Chapter 3, Chapter 4 and Chapter 5 for target detection and classification in sensor networks. Three experimental scenarios are specified, including civilian vehicle detection, SITEX00 and SITEX02 military vehicle field demos.

Chapter 7 summarizes the breakthroughs of this work, and discusses possible future development.

Chapter 2

Sensing and Acoustic/Seismic Signal Processing

As discussed in Chapter 1, a sensor node is an integrated entity that combines sensing, processing, and wireless communication capabilities. It can be equipped with different sensing modalities to capture information of the physical environment. The sensing devices are able to monitor a wide variety of ambient conditions: temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and so on [45]. In current sensor products, commonly employed sensing modalities include acoustic, seismic/acceleration, electro-magnetic waves (such as optical, infra-red), magnetic fields, images, etc.

In this chapter, we first go over the sensing mechanisms for different sensing modalities, including the devices commonly used, the physical characteristics of captured signals, and the existing problems. Due to the simplicity, low cost, and unique signal characteristics, acoustic and seismic emanations have been given considerable attention in military surveillance and environmental monitoring. In the end of this chapter, we discuss in more detail the acoustic and seismic sensing modalities in target detection and classification, and review the approaches of

acoustic/seismic feature extraction and classification in literature.

2.1 Mechanisms of Different Sensing Modalities

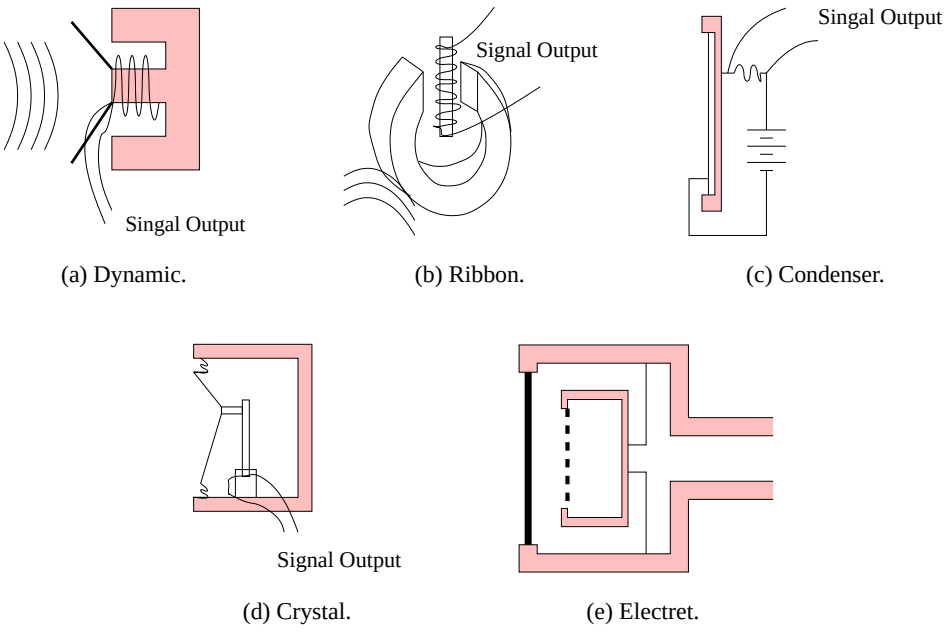
2.1.1 Acoustic Sensing

Acoustic (sound) waves typically originate from various vibrating elements such as vibrating strings or air columns by creating disturbances on the surrounding air and then expending outward. For example, the acoustic emanations of moving ground vehicles are generated by the dynamic pressure differentials occurring at the target. Sources of such pressure differentials include exhausted air and vibrating components [67]. According to the discussion in [128], the acoustic emanations from the target propagate as compression waves with a normal sound speed of 330 m/sec, and potentially impinge upon the acoustic transducer at a sensor system. The acoustic transducer, commonly referred to as a microphone, produces a voltage that is proportional to the incident dynamic acoustic pressure.

Acoustic Sensing Devices - Microphones

Microphones are used to detect acoustic signals and produce an electrical representation of the sound, i.e., a voltage or a current proportional to the sound signal. The most common microphones include dynamic, ribbon, condenser, crystal, and electret condenser microphones, which are shown in Fig. 2.1.

A dynamic microphone takes advantage of electromagnetic effects in the sense that when a sound wave hits the diaphragm (usually a thin plastic attached directly to a dense coil of wire), it moves the cone and the attached coil of wire in the field of a magnet. According to the generator effect, the movement will in turn produce a current in the wire which maps the sound pressure variation. Dynamic microphones are relatively cheap and rugged, but it has uniform responses to different frequencies which makes it unsuitable for practical usage.



(a) Dynamic. (b) Ribbon. (c) Condenser.

(d) Crystal. (e) Electret.

Figure 2.1: Different types of microphones [90].

In a ribbon microphone, a thin ribbon is suspended in a magnetic field. Sound waves move the ribbon in the magnetic field and generate a voltage between the ends of the ribbon which is proportional to the velocity of the ribbon movement [59]. Even though ribbon microphones have the advantage of discriminating against distant low frequency noise in its most common gradient form, they are rarely in use due to their sensitivity to wind noise, shock, and large sound volumes [90].

A condenser microphone is essentially a capacitor, with a thin metallic membrane and a stationary back plate charged. When a sound wave reaches the microphone, it changes the spacing between the two plates which in turn changes the capacitance of the capacitor and forces a current through the connected resistance. Compared to other types of microphones, condenser microphones have the best overall frequency response. However, the drawbacks lie in their structure of using an additional battery or external power supply to bias the plates capacitance.

Crystal microphones make usage of the piezoelectric effect of crystals that certain crystals change their electrical properties and produce voltages as they change shape. Usually, a crystal microphone uses a thin strip of piezoelectric material attached to a diaphragm. When a sound wave hits the diaphragm, it correspondingly deflects the crystal and produces a voltage proportional to the amount of deformation. The electric output of crystal microphones is comparatively large, but their frequency response is not comparable to other types such as dynamic or condenser microphones.

Electret microphones are a variant of condenser microphones that use permanently polarized electret material for their diaphragms to avoid the necessity for the biasing DC voltage in the conventional condenser microphones [90]. This kind of microphones tend to be small, cheap, and light. They are especially responsive to the range of sounds from the lower middle to the highest frequencies.

Besides the variety of basic mechanisms, microphones can also be designed with different

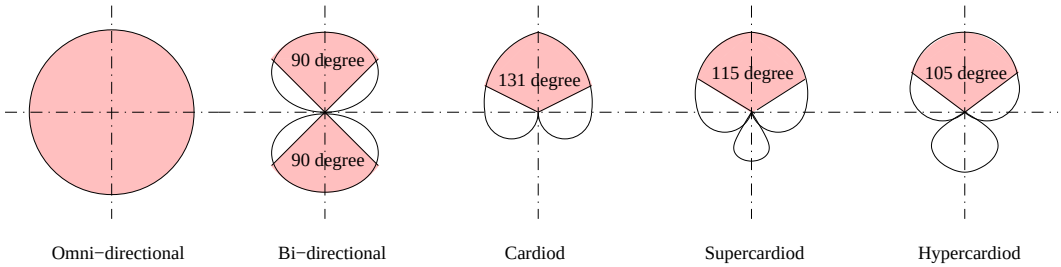


Figure 2.2: Directional patterns of microphones [90].

directional patterns and different impedances [90]. The directional patterns of microphones can be categorized into three main classes by the arc range within which they pick up at least half of the peak value: omni-directional, Cardiod, and bi-directional. Figure 2.2 illustrates the different directional patterns of microphones. Omni-directional pattern is the simplest pattern design which implies equal sensitivity in all directions. Bi-directional pattern indicates another extreme case that the microphone accepts sound striking both the front and the rear of the diaphragm and does not respond to sound from the sides (i.e., the arc of picking up is $90°$). Cardioid pattern is heart shaped that is produced by adding bipolar and omni patterns together. The basic concept of cardioid pattern is that a microphone picks up sounds in the direction it points to. According to the different arc range they pick up, cardioid pattern can be further divided into Cardiod (arc is $131°$), supercardiod (arc is $115°$), and hypercardiod (arc is $105°$).

In general, omni-directional condenser microphones are appropriate transducer choices for acoustic sensors since they have the best overall frequency response compared to other types of microphone technologies [128]. These transducers provide excellent sensitivity over a frequency range of 20Hz to $10,000$Hz. When coupled with low-noise pre-amplifiers, these kind of microphones can provide a dynamic range in excess of 120dB. In practice, however, the noise-level of these microphones is dominated by background acoustic sources, especially the wind noise. For example, a 10km/hr wind will result in a 30dB increase in the background noise relative to quiescent conditions [132]. Therefore, wind noise can significantly reduce the

Figure 2.3: An example microphone used in WINS NG 2.0 sensor platform [12].

performance of an acoustic ground sensor system.

Figure 2.3 shows the photo of a microphone integrated onto each WINS NG 2.0 sensor platform [12] developed by Sensoria Corporation to capture the acoustic signals.

Acoustic Ground Sensing Phenomenology

The phenomenology of acoustic ground sensing is significantly influenced by both atmospheric and terrain variations [128]. In the ideal case of an isotropic atmosphere and flat terrain, an acoustic wave will follow a straight path between the target and the sensor. However, in the more general case, atmospheric and terrain effects will result in reflection, refraction, scattering, and attenuation which are illustrated in Fig. 2.4. The reflection of sound follows the so-called "law of reflection", i.e., the angle of incidence equals the angle of reflection. Refraction refers to the bending of waves when they enter a medium where their speed is different. Scattering is the process that sound waves are scattered into all directions when they reach an obstacle, and attenuation is a formal effect when sound waves are traveling through a medium other than vacuum. These phenomenology effects significantly change the acoustic signal level at the sensors.

Another practical consideration of acoustic sensing is the frequency-dependent and range-

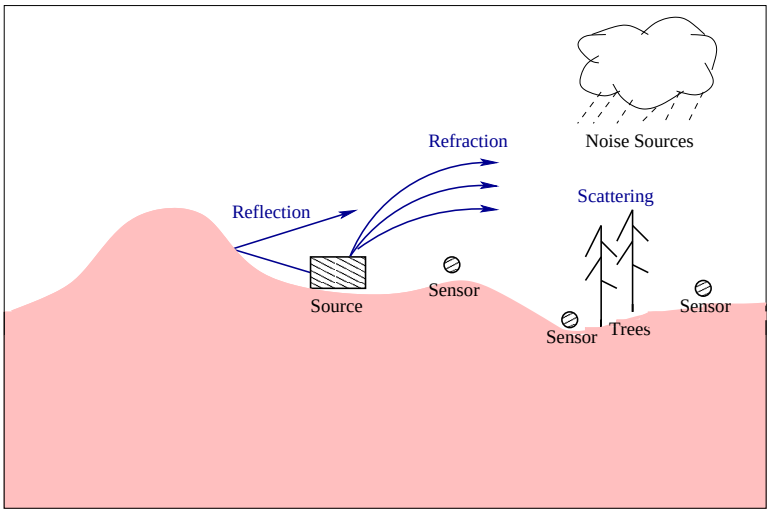


Figure 2.4: Acoustic sensing phenomenology.

dependent propagation effects, i.e., acoustic absorption, scattering, and spherical spreading result in a decrease in signal amplitude at both increasing ranges and increasing frequencies [128]. Therefore, the ability of an acoustic sensor to detect and identify a target is actually a function of its emanation frequency and the range between the target and the sensor.

The acoustic emanations from a moving vehicle can fluctuate greatly over time and apparent position, even in the absence of background noise [94]. Figure 2.5 shows a 1-second sample of acoustic signal with a sampling rate of 1024Hz and its corresponding power spectral density (PSD), in which the dynamics of the signal can be seen clearly. Target orientation, target state and multipath effects all contribute to these dynamic phenomena. The dynamics of the acoustic target signature also make it difficult to detect and identify the target of interest correctly.

Doppler Effect of Acoustic Propagation

Acoustic waves have another very important property, the so-called Doppler effect, which makes the acoustic sensing process more difficult and unique to other sensing forms. The Doppler effect was first stated by Austrian physicist Christian Doppler in 1842. The idea of

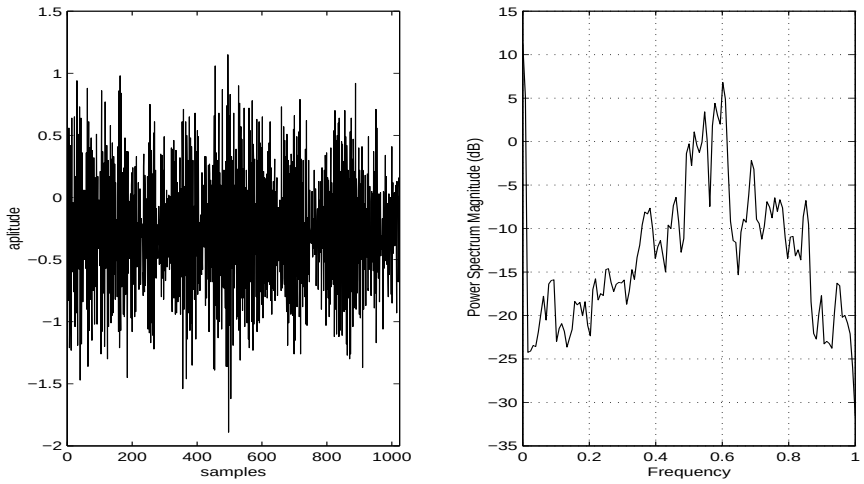


Figure 2.5: A 1-second sample of acoustic signal and its corresponding PSD.

the Doppler effect is that the frequency of a wave observed at a receiver changes whenever the source or the receiver is moving relative to each other or to the carrier of the wave, the medium [65]. If the source of waves and the receiver of waves are both relatively stationary, the receiver will detect the same frequency as the source emits. However, if they are approaching, the frequency observed would be higher; if they are diverging, the frequency would be lower. Figure 2.6 depicts the principle of the Doppler effect [41].

The left part of Fig. 2.6 shows the acoustic emanations from a stationary source of frequency f_{source}. In this case, the waves propagate in an isotropic manner and the wavelengths of the waves λ are fixed in all the directions. Suppose the sound velocity is v, then $\lambda = vT$ in period T. However, in the scenario described in the right part of Fig. 2.6, an object moves toward right at speed v_s and emanates sound of the same frequency f_{source}. The movement of the source alters the wavelength and thus the received frequency of sound, even though source frequency and wave velocity are unchanged. Considering the source speed, the wavelength of sound behind the object in period T is $\lambda^{'} = (v + v_s)T$, and the frequency is $f^{'} = \frac{v}{\lambda^{'}} = \frac{v}{v+v_s} f_{source}$. On the other side, the sound wavelength in front of the object in period T is $\lambda^{''} = (v - v_s)T$,

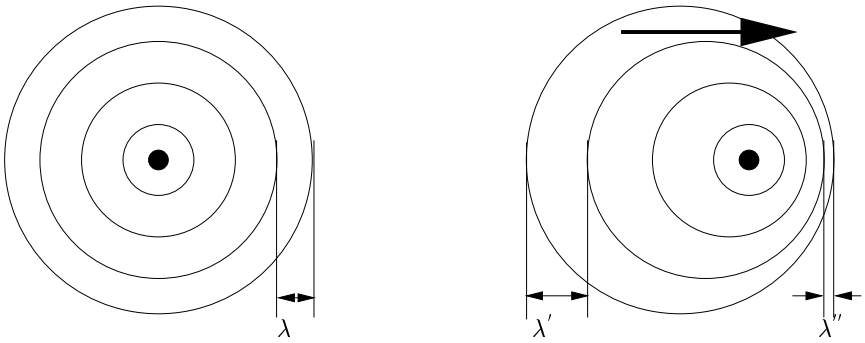


Figure 2.6: The principle of the Doppler effect.

and the frequency is $f'' = \frac{v}{\lambda''} = \frac{v}{v - v_s} f_{source}$.

Considering the acoustic sensing phenomenology discussed, both the dynamics of the acoustic signals and the Doppler effect propose additional burdens on the moving vehicle detection and identification using acoustic signals.

2.1.2 Seismic/Acceleration Sensing

For moving ground vehicles, seismic emanations occur when vibrating components within the target are mechanically coupled to the ground. Target vibrations that are coupled into the earth propagate within the ground as elastic waves. These elastic waves propagate in a complex manner through the geologic medium, and potentially impinge upon the seismic transducer of the ground sensor system [128].

Seismic Sensing Devices - Geophones/Accelerometers

Seismic transducers measure the earth's vibrations at a single point. There are two types of seismic transducers commonly used within a ground sensor system. The first type is referred to as a geophone, which is a moving-coil electro-dynamic device that produces a voltage proportional to the velocity of the ground vibrations [128]. The structure of a geophone is illustrated

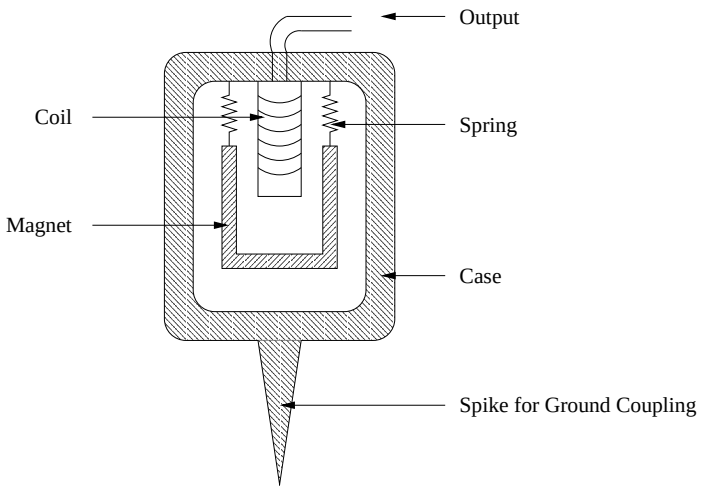


Figure 2.7: The structure of a geophone.

in Fig. 2.7 [138].

When seismic waves arrive at the detection point, the geophone case with rigidly attached coil is coupled to the ground and thus moves harmoniously. The magnet that is suspended on the springs remains effectively stationary because of its inertia. The coil movement within the magnetic field induces an electrical voltage across the coil which is proportional to the relative velocity of the coil and the magnet [89]. Therefore, the response of a geophone is proportional to the ground velocity as it depends on the relative velocity of the magnet and the coil [42]. Figure 2.8 shows the photos of example geophones used in ground sensor systems and the quarter coin serves as a reference of scale. Different styles of geophone cases can be implemented in different environments. For example, the rightmost geophone shown in Fig. 2.8 (the one without spike) is designed for use on hard surfaces into which spikes can not be pushed.

Geophones exhibit both a low-frequency fundamental resonance and a high-frequency spurious resonance. As discussed in [128], manufacturers meticulously specify the fundamental resonant frequency of commercial geophones, that are common set to be 10Hz, 14Hz, 28Hz, and 40Hz. The spurious resonance usually occurs at 25-times the fundamental resonance, and

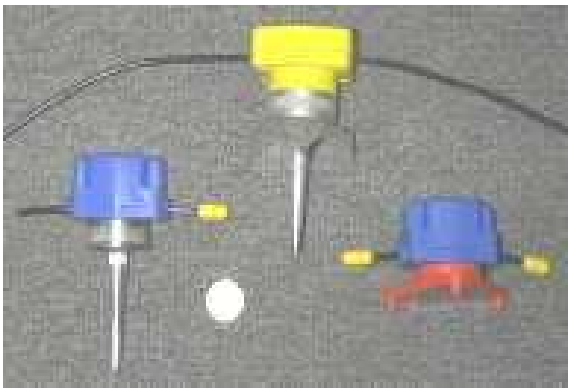


Figure 2.8: Three example geophones used in ground sensor systems [48]. (The quarter coin in the figure serves as a reference of scale.)

is often specified by the manufacturer. Because of amplitude and phase distortions at the fundamental and spurious resonant frequencies, ground sensor systems should generally exploit only those frequencies between the fundamental and spurious resonances. For example, when using a 14Hz geophone, the recommended seismic bandwidth is from 15Hz to 340Hz. For wider bandwidth systems, another type of seismometers, accelerometers, are often considered.

Accelerometers are solid-state piezoelectric devices that produce a voltage proportional to the acceleration of the ground vibrations. Accelerometers exhibit a high-frequency resonance which is typically in excess of 1000Hz with no low-frequency resonance. Hence, the seismic bandwidth for accelerometers is between 0Hz and 1000Hz. Despite the frequency bandwidth advantage, current accelerometers are fragile and exhibit excessive electronic noise at low frequencies which limit their usefulness. It has been shown that geophones are preferred for sensing frequencies below approximately 100Hz, while accelerometers are preferred for sensing frequencies above 100Hz [127]. In many ground sensor applications like in moving vehicle detection, the majority of propagated seismic energy is below 100Hz, and thus geophones are most popularly used [128].

Seismic Signal Characteristics

Seismic waves are elastic waves propagating through the earth. Both natural and human-made sources of deformational energy can produce seismic waves. They have been deployed in many applications, such as the study of earthquake, modern exploration for oil and gas, etc. There are several different kinds of seismic waves, and they all move in different ways. In general, seismic waves can be classified into two broad categories: body waves that travel through the earth's inner layers at a higher speed and propagate in three dimensions; and surface waves that can only move along the surface of the ground and propagate in two dimensions [138].

The body waves can be sub-categorized into two types: compression (P) waves and shear (S) waves. P waves are the fastest kind of seismic waves. In P waves, particles constituting the medium are displaced in the same direction that the wave propagates, in this case, the radial direction [48]. Thus, material is being extended and compressed as P waves propagate through the medium. Whereas, in S waves, particles constituting the medium are displaced in a direction that is perpendicular to the direction that the wave is propagating.

Surface waves propagate at speeds that are slower than body waves since they are less efficiently generated by buried sources and have amplitudes that decay with distance from the source more slowly than is observed for body waves [48]. As body waves, surface waves can also be sub-categorized into two types: Love waves and Rayleigh waves. Love waves is named after A. Love who worked out the mathematical model for this kind of wave in 1911. It is the fastest surface wave and moves the ground from side-to-side. Rayleigh waves are predicted in 1885 by L. Rayleigh. A Rayleigh wave moves the ground up and down, and side-to-side in the same direction that the wave is moving [121]. The schematics of all the four kinds of seismic waves are illustrated in Fig. 2.9. The arrows in the figure show the direction that the waves are moving. The relationship between the propagation direction and the particle motion is shown clearly in the figure.

Seismic wave propagation is highly dependent on the underlying geology. As illustrated in

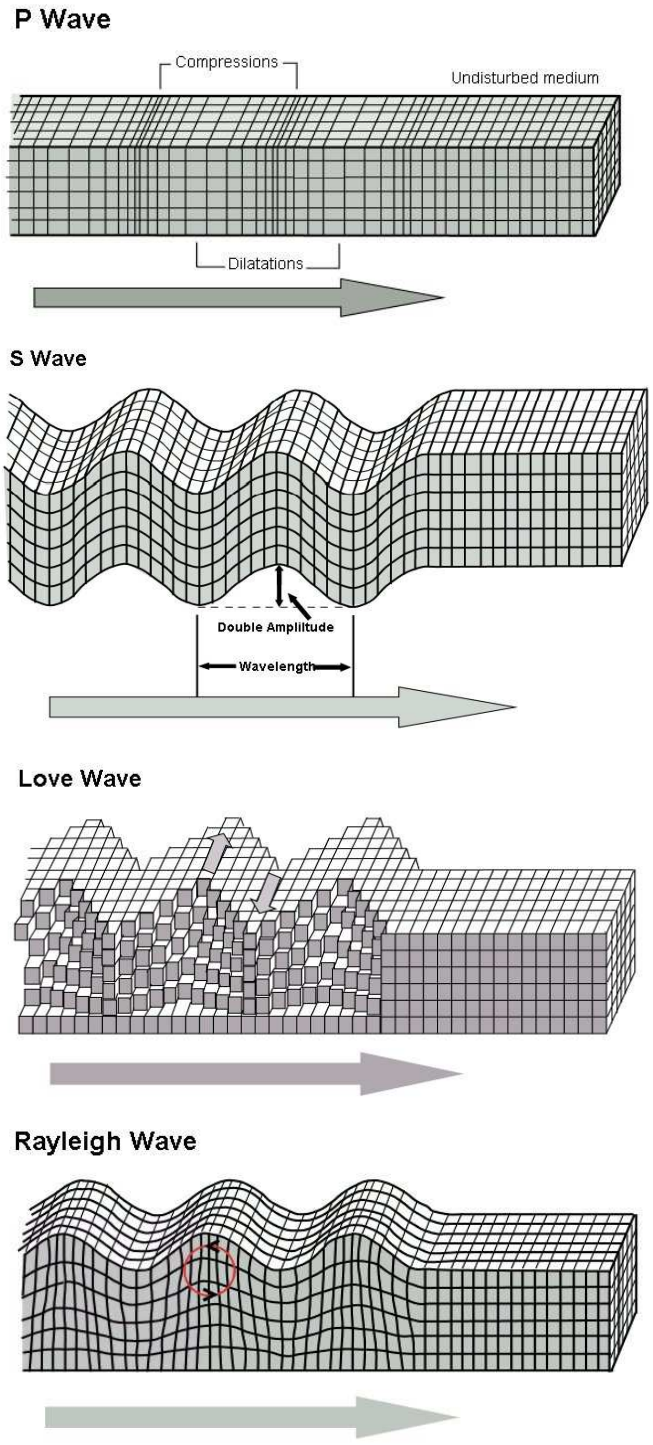


Figure 2.9: Schematics of four kinds of seismic waves. Courtesy of Dr. W. Pennington [121].

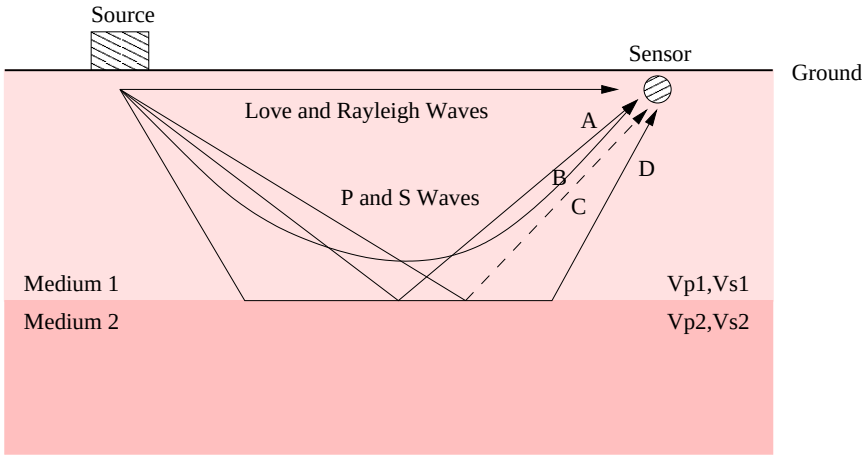


Figure 2.10: Propagation paths of seismic waves (redrawn from [30]).

Fig. 2.10, different kinds of seismic waves have various forms of propagation paths [30]. When the vibrations generated by the source propagate through the earth, the wave field becomes very complicated. Love and Rayleigh waves travel along the surface, while P and S waves travel through the deeper parts of the earth in multiple forms, one form of "multipath" occurs in the direct path (B) from the source to the sensor, other forms, such as reflections (A) and refractions (D) also arise. When seismic media reflection occurs, the mode conversion are also generated, which is shown by path C where P waves (solid line) convert to S waves (dashed line). S to P conversions (not shown) can also occur at the interface between medium 1 and medium 2 [30, 138].

The amplitude of any seismic wave is dependent on numerous factors including source coupling, receiver coupling, source-receiver distance and the properties of the material through which the seismic wave propagates [44]. Figure 2.11 shows a 1-second sample of observed seismic signal from a moving ground vehicle and its corresponding PSD.

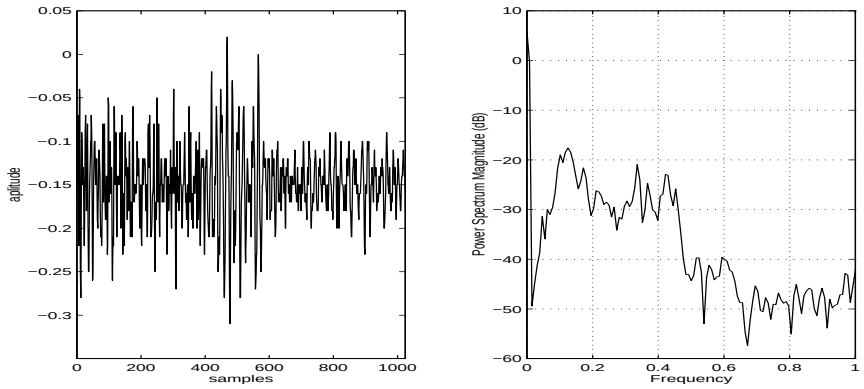


Figure 2.11: A 1-second sample of seismic signal and its corresponding PSD.

2.1.3 Infra-red Sensing

Infra-red (IR) radiation is electromagnetic wave of wavelength just greater than that of red visible light. IR waves can be simply considered as heat, because they're given off by hot objects, and they can be felt by human beings as warmth on the skin. Actually, all objects with temperatures above absolute zero (0K, i.e., $-273\,^{\circ}\mathrm{C}$) contain kinetic heat due to molecular motions. The concentration of this kind of heat can emanate into the surrounding environment and be measured by a thermometer. This process is known as Infra-red sensing, also known as heat sensing or thermography.

IR waves are used for many tasks, for example, remote controls for TVs and video recorders, and heat lamps used by physiotherapists to help heal sports injuries. Since every object gives off IR waves, they can be used to "see in the dark". A sensitive IR detector is sometimes used for night sights of weapons. Apart from remote controls, one of the most common modern usage of IR is in the field of security. Infra-Red motion detectors are used in alarm systems to pick up the IR radiations emitted by moving objects such as personnel, vehicles, etc. that have detectably different temperatures than the surrounding ambient temperature.

According to the motivator of IR detection, IR motion detectors can be divided into two

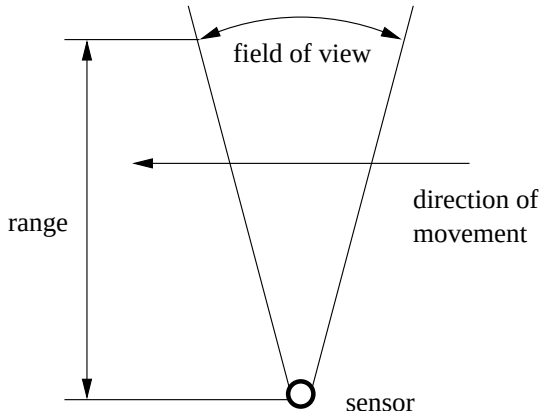
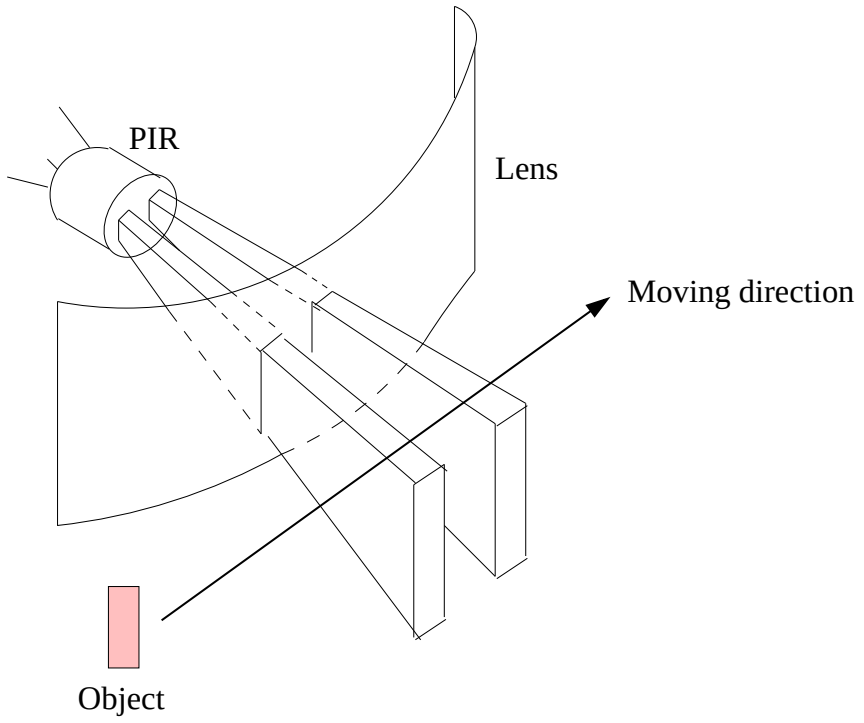


Figure 2.12: Sensor coverage of PIR motion detectors.

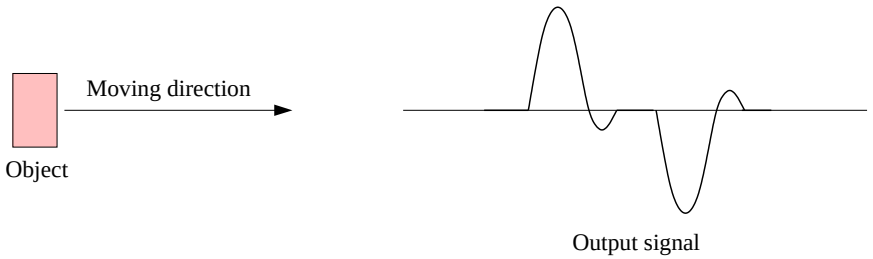
main categories: passive infra-red (PIR) and active infra-red (AIR) motion detectors.

Passive Infra-red Motion Detectors

In a PIR motion detector, a sensor containing an infra-red sensitive phototransistor is placed in the area to be monitored. A lens in front of the phototransistor allows the sensor to divide an area into several distinct zones within the range of the detector. These zones are approximately conical shaped, fanning out from the lens across the monitored area and down to the floor, as illustrated in Fig. 2.12 [135]. Figure 2.13(a) shows the structure of an example PIR motion detector (PIR325) which has two sensing elements connected in a voltage bucking configuration. A target passing in front of the sensor will activate the first one and then the other element whereas other sources (such as vibration, temperature changes, etc.) will affect both elements simultaneously and be canceled [50]. Therefore, for a target to be detected, it must pass into or out of one of the coverage zones. Targets moving directly toward or away from the sensor may sometimes remain undetected, which is more likely to occur for small objects. For best detection performance, targets should move across the sensor coverage as indicated by the arrow in Fig. 2.12. Generally speaking, target size, distance from sensor, and surface temperature



(a) Structure of PIR325.



(b) Output signal.

Figure 2.13: Structure of example PIR motion detector (PIR325) and the output signal [50].

Figure 2.14: CLIP series of PIRs. Courtesy of Visonic Ltd.

all play important roles in the sensing performance of PIR motion detectors. When the sensor detects movement, it will generate an electrical signal whose level varies depending on the size, temperature, speed, and distance of the target that is detected. The source movement and the corresponding output signal from the sensor are shown in Fig. 2.13(b).

PIR motion detectors are widely used due to their advantages including: 1) ease and neatness of installation; 2) low cost; 3) no need for units to interact with each other; 4) reliability; and 5) resistance to rattling or vibrating items. However, they can be subject to false alarms under certain conditions, for example, where there may be sudden fluctuations in temperature or warm air movement. Figure 2.14 shows the CLIP series of PIR motion detectors developed by the Visonic LTD. as an example.

Active Infra-red Motion Detectors

Unlike PIR sensors, AIR motion detectors consist of two units, one containing an infra-red photodiode and an infra-red sensitive phototransistor, the other containing an infra-red reflector. When deployed in the area to be monitored, it will project an infra-red beam from the phototransistor and the receivers, as shown in Fig. 2.15. A target who moves through the beam breaks a circuit in the receiver, causing an output from the sensor [136].

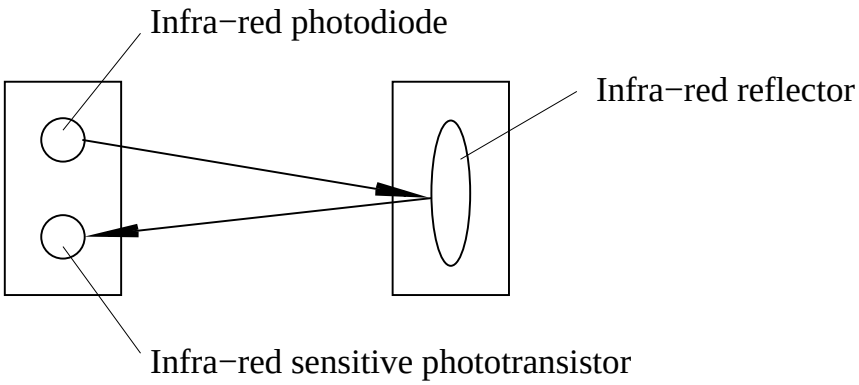


Figure 2.15: Principle of AIR motion detector.

2.1.4 Optical Sensing

Optical sensing refers to the methods that translate the information occurring as variations in the intensity or some other properties of light into an electrical signal. In this section, we constrain the discussion of optical detectors to the wavelength range of visible lights. There are different types of optical detectors that are sensitive to different wavelength, including the charge-coupled detector array (CCD), the semiconductors, the photodiode array (PDA), and the photomultiplier tube (PMT), etc. They have been used in a variety of applications, including medical instrumentation, encoders, position sensing, fiber-optic communication systems, and image processing. The main advantages of optical sensing include remote operation, extremely high sensitivity, small dimensions, low power consumption.

A CCD is an integrated-circuit chip that contains an array of metal-oxide-semiconductor (MOS) capacitors. Each CCD cell (MOS capacitor) is properly biased so that when lights hit the device, it can create electron-hole pairs within the silicon and form an electron potential well as shown in Fig. 2.16. Through this photoelectric effect, the capacitors can accumulate and store charge in response to the amount of light they receive. By clocking the gate voltages, the array charge can read out by transferring the charges of neighboring capacitors in a serial fashion.

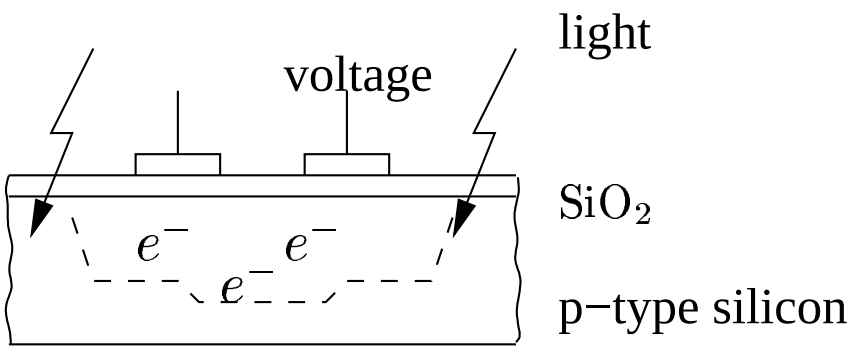


Figure 2.16: The schematic of a CCD cell.

CCDs can be used in most optical sensing applications. Compared to other array detectors, it presents a much higher sensitivity to low-intensity-level lights.

Semiconductor-based optical detectors are also based on the photoelectric effect to generate electron-hole pairs corresponding to the amount of light received by the semiconductor. Photodiodes are typical devices in this category. Photodiodes can be made with any semiconductor materials in which a p-n junction is formed. There are several different kinds of photodiodes, but they all work on the same principle which is based on photoconductivity. In photodiodes, a voltage bias is present and the photodiodes are reverse-biased, which means that a positive bias is applied on the n side of the diode while a negative bias is applied on the p side. When in operation, a photodiode picks up a light signal and creates an electron-hole pair. The electron and the hole are swept through the junction in opposite directions, producing a modulated current in the photodiode. The photodiode captures variations of light intensity and displays the readings to the user. For best performance, light must be in a tightly collimated beam when arriving at the photodiode; if the beam diameter is larger than the detector's, the signal contained in the diverging light will be lost [75]. Generally, photodiodes are very small, sensitive and require litter power to operate. Figure 2.17 shows several example photodiodes in use.

A PDA is simply a linear array of discrete photodiodes on an integrated-circuit chip. It

Figure 2.17: Example photodiodes in use. Courtesy of O/E Land Inc. [98].

works on the same principle as simple photodiode and is normally used to detect a range of light wavelengths simultaneously. When in operation, the charge generated by each element is read out sequentially and the final output is a function of linear distance along the array [93].

A PMT is an extremely sensitive photocell which converts light signals of a few hundred photons into a usable current pulse. It consists of a photocathode, a series of dynodes, and an anode, as shown in Fig. 2.18. The photocathode comprises a photosensitive layer that can convert the incident light photons into electrons due to the photoelectric effect. Since PMTs are usually used for very low-level light signals and the number of emitted electrons is comparable to the number of incident photons, the charge on the photoelectrons will be too small to provide a detectable electrical signal. Therefore, a series of dynodes are used as an amplifier on which a secondary-electron multiplication process occurs. This multiplication effect creates 105 to 107 electrons for each photon hitting the first cathode depending on the number of dynodes and the accelerating voltage. After amplification, the electrons are collected at the anode and a current signal is generated.

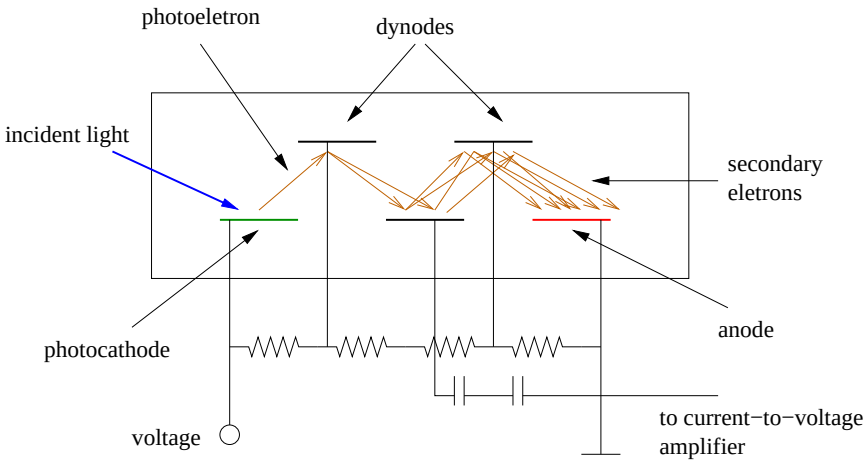


Figure 2.18: The schematic of a PMT (redrawn from [93]).

2.1.5 Magnetic Sensing

Magnetic sensors have been in use for over 2,000 years. Magnetic field sensing has vastly expanded as industry has adapted a variety of magnetic sensors to detect the presence, strength, or direction of magnetic fields not only from the earth, but also from permanent magnets, magnetized soft magnets, vehicle disturbances, brain wave activity, and fields generated from electric currents [22]. Magnetic sensors can measure these properties without physical contact and have become the sensing devices of many industrial and navigation control systems. The technology for sensing magnetic fields has also evolved driven by the need for improved sensitivity, smaller size, and compatibility with electronic systems.

Magnetic sensors can only indirectly detect parameters like the presence, strength, or direction of magnetic fields. First, the enacting input has to create or modify a magnetic field, for example, a large ferrous object such as automobiles, and airplanes moving within the earth's magnetic field can create this kind of field variation. Once the magnetic sensor detects the field variation, the output signal requires some signal processing to translate the sensor signal into the desired parameter value [22].

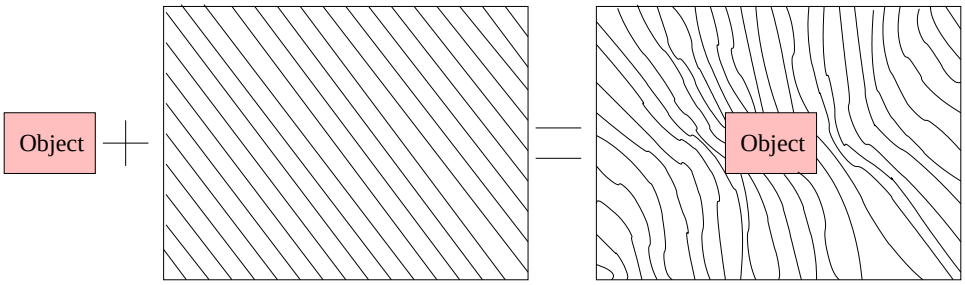


Figure 2.19: Ferrous object disturbance in uniform field [22].

According to the field sensing range, various magnetic sensors can be divided into three categories: low field, medium field (also called earth's field), and high field sensing. Since the magnetic range for the medium field sensors is comparable to the earth's magnetic field, this kind of sensors are most widely adopted to use the earth's magnetic field and detect anomalies in the earth's field for vehicle detection, or to determine compass headings for navigation, etc. For example, in vehicle detection scenarios, the earth's field provides a uniform magnetic field over a wide area. A ferrous object, such as a ground vehicle, a plane, or a train creates a local disturbance in the field whether it is moving or remain still, as illustrated in Fig. 2.19. An example magnetic sensor (DT028 from the Logical Interface Online) used to detect the change in the earth's filed is shown in Fig. 2.20.

2.2 Acoustic/Seismic Signal Processing

A moving ground vehicle can be detected in many distinct ways, for example, the engine, exhaust and other vibrating components of the target can emanate acoustic energy, the rolling tires that contact the roadbed can generate seismic vibrations in the ground, and it can also alter the thermographical pattern when passing by PIRs. The acoustic and seismic signals exhibit unique signatures that are sufficient to identify different targets, and the microphones or geophones are comparatively cheaper than other sensors which makes it possible for large amount

Figure 2.20: An example magnetic sensor (DT028). Courtesy of the Logical Interface Online.

of deployment in wireless sensor network applications. Therefore, acoustic and seismic sensing modalities are widely used in civilian and military areas. The acoustic/seismic signal processing system consists of two stages: feature extraction and pattern classification. In this section, we review the available signal processing techniques for acoustic and seismic signals.

2.2.1 Feature Extraction

Feature extraction is the process to obtain signal characteristics from the time series data. It can be considered as a data compression process which removes irrelative information and preserve relevant one from the raw data [138]. Feature extraction plays an important role in target classification problem since the performance of the classifier largely depends on the quality of the feature vectors. In order to conquer the non-stationarity of the captured signal from sensors and derive robust features, researchers have developed feature extraction algorithms in three domains: time domain, frequency domain, and time-frequency domain.

Signal processing in the time domain is the most natural approach, but due to the complexity of environment, it is not an optimal solution to the problem. In [119], Sampan used a block-averaging method on the time domain signals to classify vehicles into 4 different classes by using fuzzy neural network architecture. Basically, the method is based on the short time strength of the signals, and little spectrum information is used. Therefore, it can not distinguish

two different vehicles with nearly the same size and the engine power. Another important algorithm to process signals in the time domain is beamforming. It is a spatial filtering operation that the signals from look direction are passed and noise from other directions is rejected by placing deep noise rejecting nulls in the directions of interfering noise sources. Braunling developed an adaptive beamforming algorithm in [17] which is theoretically optimum. However, the time domain features are not robust due to the interfering noise, the complicated waveforms, the reflection and refraction of the signals, and the variation of the environment.

The feature extraction algorithms of signals in the frequency domain are based on their spectrum which is generated by using the Fourier transform. For example, the usable frequency range for acoustic signal processing of ground vehicles is limited approximately to the range of 20Hz to 200Hz, since frequencies below 20Hz are dominated by wind noise, and those above 200Hz are dominated by poor propagation characteristics. In this low end of spectrum, vehicles have two main sources of sound: the engine and the propulsion gear that are periodic.

Based on this knowledge, Wellman described the harmonic line association (HLA) algorithm for acoustic signal feature extraction, and the seismic shape feature extraction in [144]. The HLA algorithm takes the advantage of spectral characteristics that are dominated by narrow band spectral peaks. It selects peaks that are harmonically related to create harmonic line sets for each frame of data samples. Then a split-window peak-picking method is implemented on the power spectral estimates (PSE) feature set to find the maximum peak in the frequency domain. These peaks are assumed as some kth harmonic lines of the fundamental frequency. After calculating the total signal strength in the HLA set, the integer k is assumed to be the correct harmonic line number, and the harmonic lines of this particular set are retained as a feature vector. The shape analysis of seismic signals uses the higher order shape statistics of seismic signals to form the feature set, which includes the mean, the standard deviation, the skewness and the kurtosis of signals. The combined algorithm can remove the effect of engine speed in some sense and the effect of distance can be partially removed by using the relative harmonic

amplitude [76, 133].

Another feature that is commonly used to represent signals in the frequency domain is their corresponding power spectral density (PSD). The PSD of a signal is the Fourier transform of its autocorrelation function. It tells the energy distribution of acoustic signals along their frequency ranges.

As we have mentioned, one of the most significant characteristic of ground vehicle emanated signals is that they are non-stationary and wideband, which makes it very difficult to pick peaks in the frequency spectrum. Under this situation, approaches to extract features in time-frequency domain are developed. The most popular time-frequency feature extraction method is the wavelet transform, and it exhibits great performance in this area of application [25, 86].

Choe used combined short time Fourier transform (STFT) and discrete wavelet multi-resolution analysis to extract features in [27]. After the analog-to-digital converter and downsampling the signals to 8 or 16KHz, a real-time wavelet transform algorithm is implemented on the digital signals and some statistical parameters are calculated, such as the first-order parameters and the energy content of the signals. Using neural network classification method, he got a 98.25% accuracy for a two-class vehicle classification problem.

Averbuch developed a Discriminant Block Pursuit algorithm which is based on the assumption that the signature for each class of signals is obtained as a combination of the inherent energies in a small set of the most discriminant blocks of the wavelet packet coefficients [9]. Therefore, the signature of a certain vehicle can be constructed using the distribution of the energies among blocks which consist of wavelet packet coefficients. Using this kind of feature extraction method, the authors presents a solution of both the classification and the detection problem of a moving vehicle.

2.2.2 Pattern Classification

Once features are extracted from the time series data, classification process can be executed on the extracted feature vectors. Basically, the goal of pattern classification is to partition the feature space into regions following a specific metric. Every region corresponds to one category of the targets. Generally speaking, the available pattern classification algorithms can be divided into two classes: supervised learning and unsupervised learning.

Prior to the learning process, if the learner is given a set of examples which is called a training set and each example shows what output will be returned for a given input then this type of learning can be classified as supervised learning. Many algorithms have been developed in this category including Bayesian approach, nearest neighbor rule, decision trees, neural networks, and Hidden Markov Models. In contrast to supervised learning, if the learner is not presented with an explicit set of examples showing what kind of desired input/output relations there should be, then this type of learning is an unsupervised learning. In other words, there is no target specified for the learning process, and the learning process is mainly based on the input data and self-organization. The common classification algorithms based on unsupervised learning rule includes k-means clustering, and self-organizing maps (SOM) approach.

In [80], the authors compared the performance of three supervised classification algorithms on acoustic and seismic signals: k-nearest neighbor (kNN) classifier, maximum likelihood (ML) classifier using Gaussian data modeling, and support vector machine (SVM) classifier. Given a set of N-dimensional feature vectors $\{\mathbf{x}; \mathbf{x} \in R^N\}$. In supervised learning, it is assumed that each of the feature vectors is assigned a class label, $\omega_c \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$, that belongs to a set of m elements. The prior probability that a feature vector belongs to class ω_c is denoted by $p(\omega_c)$, and similarly the posterior probability for class ω_c given that $\mathbf{x}$ is observed is denoted by $p(\omega_c|\mathbf{x})$.

The kNN classifier is a simple supervised classification algorithm which deploys the whole training set. The basic idea of kNN is to classify a test sample by assigning it to the class

label most frequently represented among the k nearest neighbors. During the test process, the distance between each test vector and every training vector (prototype) is calculated and the k closest prototype vectors in the training set from the test sample are retrieved. If within this neighborhood, more samples lie in class ω_i than any other classes, the unknown test sample is assigned as belonging to class ω_i. As the size of training set increases, the kNN classifier is not suitable for actual implementations since it requires too much memory and processing power for distance computation.

In the ML classifier used in [80], the distribution of the training samples of the same class is modeled as a mixture of Gaussian density functions, i.e., the likelihood function is modeled as follows:

$$p(\mathbf{x}|\omega_i) = G(\mathbf{x}|\theta_i) = \sum_k \alpha_{ik}|\Lambda_{ik}|^{-N/2} \exp(-\frac{1}{2}(\mathbf{x} - m_{ik})^T \Lambda_{ik}^{-1}(\mathbf{x} - m_{ik})) \tag{2.1}$$

where $\theta_i = [\alpha_{i1}, \ldots, \alpha_{iP}, m_{i1}, \ldots, m_{iP}, \Lambda_{i1}, \ldots, \Lambda_{iP}$ are the mixing, mean, and covariance matrix parameters of the P mixture densities corresponding to class ω_i. These parameters can be derived from the training samples of each class. The discriminant function is computed as $g_i(\mathbf{x}) = G_i(\mathbf{x}|\theta_i)p(\omega_i)$ where the prior probability $p(\omega_i)$ is assumed to be uniform distributed.

A SVM is a linear classifier operating in a higher dimensional space. It transforms the N-dimensional input vectors into a M-dimensional feature space following the transformation functions $\{\phi_i(\mathbf{x})\}_{i=1}^M$ where $M > N$. A linear classifier with corresponding weights $\{w_1, w_2, \ldots, w_M\}$ can be represented by a discriminant function $g(\mathbf{x}) = \sum_{j=1}^M w_j\phi_j(\mathbf{x}) + b$ in the M-dimensional feature space and b is the bias parameter of the classifier. Choosing a subset of training vectors, which are called the support vectors, the optimal weight vectors can be represented as $w_j = \sum_{i=1}^Q \alpha_i\phi_j(\mathbf{x}_i), j = 1, 2, \ldots, M$ and the discriminant function of the classifier becomes $g(\mathbf{x}) = \sum_{i=1}^Q \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$, where $K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^M \phi_j(\mathbf{x})\phi_j(\mathbf{x}_i)$ is the symmetric kernel of the SVM. Therefore, the SVM can be computed using the kernel representation and the support vectors.

In [80], three-way cross validation is used to assess the performance of the three classifiers on real acoustic and seismic signals collected in the SITEX00 experiments. The low bandwidth seismic signals were sampled at 256Hz, and the feature vectors consist of the positive 64 samples of 128 length FFT corresponding to 0.5s time series segments. Two types of targets are considered: tracked and wheeled vehicles. The true positive rates (correct classification rates) of kNN are 89.55% for tracked and 95.2% for wheeled vehicles, while ML achieves 92.27% for tracked and 77.6% for wheeled vehicles. SVM classifier has the highest true positive rates of 94.09% for tracked and 97% for wheeled vehicles. The three classifiers present similar results on the wideband acoustic signals, that is, SVM exhibits better performance than kNN and ML classifiers.

Artificial neural network (ANN) classifier derives its computational power from the parallel distributed structure and has the ability to learn and adapt dynamically. For target identification, ANN can provide both a robust classifier and a measure of confidence in the classification decisions [144]. A lot of studies have been devoted into the application of ANN in target detection and classification. For example, in [144], the authors proposed to classify vehicles using a backpropagation neural network (BPNN) with an adaptive learning rate, which allows fine-grain adjustments during training. Further refinements to the learning rate are accomplished through an interlayer multiplier, which only affects the learning rate in the hidden neuron layer. Smoothing method is also incorporated to allow the control of weight adjustment based on the past values of gradient descent and to prevent the training process from terminating in shallow local minima. The ANN classification performance is qualified through a confusion matrix that provides the percentage of correct identification for each class of targets. When performed on the feature vectors derived from the harmonic line association (HLA) of acoustic signals and the shape statistics of seismic signals, the average accuracy can achieve 80% for the classification of three target types.

SOM is a widely used unsupervised classification technique that consists of a defined num-

ber of neurons in a two-layer network. It has the advantage that the models are automatically organized into a meaningful two-dimensional order in which similar models are closer to each other in the grid than the more dissimilar ones [70]. Wolford generalized the existing SOM algorithm in [146], which presented to train SOM with competition learning. A discriminant function for each neuron is computed and the highest function value is identified, then the corresponding neuron and its neighbors are activated. An adaptive process is then implemented to strengthen the discriminant value of the activated neurons in response to the inputs [138, 146]. Due to the partial activation of neurons during training process, SOM is not computationally efficient in implementation.

Chapter 3

Target Detection in Sensor Networks

Within various applications of sensor networks, target detection is a typical one that has drawn much attention in both battlefield surveillance, civilian and environment monitoring. Different sensing modalities present various characteristics on target detection. In this work, we focus on using acoustic signals for target detection purpose. In this chapter, both single and multiple target detection problems are discussed. In general, single target detection can be implemented using a simple energy detector, while multiple target detection exhibits several challenges due to the extremely constrained resource and scalability issues. In an attempt to solve these problems, a cluster-based distributed estimation framework is developed. We assume the sensor network has been divided into clusters, with each sensor node belonging to only one cluster, then the local estimation is generated within each cluster and a fusion algorithm is performed to combine the local results. In order to eliminate raw data transmission for local estimation within each cluster, a progressive detection approach is presented based on the classic centralized Bayesian estimation algorithm and is implemented using a mobile agent framework. In contrast to the centralized scheme in which data from all the sensors need to be transmitted to a central processing unit, the progressive approach sequentially estimates the number of targets based on only the local observation and partial estimation result from the previous sensors. Then a pos-

terior probability fusion method is derived based on Bayes theorem and is compared with the Dempster's rule of combination.

3.1 Single Target Detection

Single target detection in sensor networks can be achieved using a simple energy detector, i.e., to detect a target/event when the output exceeds an adaptive threshold. The output of the detector should follow several principles [120]:

- At any instant, the detector output should be the average energy in a certain window.
- The detector output should be sampled at a certain rate based on a priori estimate of the target velocity.
- The time series data for detected event should be provided for further processing, such as classification.

In an ideal isotropic sensing field, the distance between the target and the sensor can be estimated from the energy decay model

$$x_i(t) = \frac{s(t)}{d^{\alpha}} \tag{3.1}$$

where $x_i(t)$ denotes the ith sensor energy reading at time t, $s(t)$ is the energy emitted by the target, d is the distance between the target and the ith sensor, and α is a known constant which depends on the physical characteristics of the environment. Normally, we can define $\alpha = 2$.

3.2 Multiple Target Detection and Separation

Different from the situation of single target detection, multiple target detection in sensor networks remains a challenging problem due to the non-stationarity and correlations of signals. On

top of that, the limited wireless communication bandwidth, the large amount of sensor nodes, and the energy efficiency requirement of sensor networks all present further difficulty to the multiple target detection problem. Researchers have constructed approaches using different sensing modalities, 1-D or 2-D, to infer this problem. For example, images are widely used media in sensor networks for multiple target detection task. Through image segmentation, multiple targets can be easily separated from the background. However, if multiple targets are overlapped from the view that the image is captured, it is very difficult to distinguish them from each other using traditional segmentation method, even though they may be far away in geographical positions. In this kind of situations, 1-D signals such as acoustic and seismic signals are preferred because of their intrinsic relationship among the captured signals from sensor nodes, the target signatures, and their relative positions. For example, in the case of multiple target detection using acoustic signals, especially when the targets are close to each other, the acoustic signal captured by individual sensor node is a linear/nonlinear weighted combination of the signals emitted from different targets with the weights determined by the signal propagation model and the distance between the target and the sensor node. In this chapter, we focus on the problem of multiple target detection in sensor networks using acoustic signals.

If we consider different targets in the field as sources and assume the signals they generate to be independent, multiple target detection in sensor networks can be solved using traditional blind source separation (BSS) technique where the signal captured by individual sensor is a linear/nonlinear weighted mixture of the signals generated by the sources and independent component analysis (ICA) algorithm is a widely accepted technique to solve BSS problem. The "blind" qualification in BSS refers to the fact that there is no *a priori* information available on the number of sources, the probabilistic distribution of source signals, or the mixing model [13, 58, 137]. However, for conceptual and computational simplicity, the majority of BSS algorithms employ the linear, instantaneous mixture model and make the assumption that *the number of sources equals the number of sensors* so that the mixing/unmixing matrix is square

and easy to be estimated.

Although the equality assumption is reasonable in small-size, well-controlled sensor array processing, it is not the case in sensor networks where thousands of sensor nodes are usually deployed densely within the sensing field and the number of sensors is far more than the number of sources. Therefore, there is an immediate need to estimate the number of sources, a problem also referred to as *source number estimation* or *model order estimation* in [114]. Several techniques have been proposed trying to tackle this problem, some heuristic, others based on more principled approaches. As discussed in [114], techniques of the latter category are clearly superior and heuristic methods at best may be seen as approximations to a more detailed underlying principle. Some examples of principled estimation approaches include the Bayesian estimation method [115], Markov chain Monte Carlo (MCMC) method [113], and variational learning approximation [8].

The source number estimation approaches mentioned above have been successfully implemented in several applications like music mixture separation and EEG (human brain activity) detection [114, 115]. However, their usage in sensor networks is hindered by their *centralized structure* where the raw data from a large amount of sensors have to be transmitted to a central processing unit, which will generate significant data traffic over the network, occupy communication bandwidth, and consume energy.

In order to reduce the amount of data transmission and conserve energy, we presented a cluster-based distributed source number estimation algorithm in [140] where sensor networks are divided into clusters of small number of sensors and centralized source number estimation is performed within each cluster using the Bayesian estimation approach. The estimated posterior probabilities of possible source number hypotheses from different clusters are then combined using a probability fusion rule. The cluster-based estimation approach can reduce the amount of data transmission to some extent. However, the centralized processing within each cluster still requires raw data transmission and in turn consumes certain amount of network bandwidth

and energy. Moreover, within each cluster, the centralized estimation has to be performed at a central processing unit (the cluster head), which would place the burden of computation on the cluster head. This is contradictory to the property of sensor networks that all sensor nodes should have similar functionality and they should take turn to be elected as the cluster head [56].

In order to *completely* eliminate raw data transmission in solving the multiple target detection problem, we present a progressive approach in this section and integrate it with the cluster-based distributed source number estimation scheme we developed. The progressive approach carries out the estimation process within each cluster in a sequential manner. Instead of sending data from all sensors to a central unit as in the centralized scheme, the determination of possible number of targets at a sensor is only based on its local observation and the estimation result received from its previous sensor. Therefore, raw data transmission is avoided and only small packets of partial estimation results are transmitted through the networks. The progressive approach is developed based on the Bayesian estimation method since the Bayesian framework has a solid statistical basis and it also accommodates the usage of Bayesian probability fusion rule in the distributed scheme.

3.2.1 Terminologies and Problem Definition

To make our presentation more clear, we first define several terminologies and symbols used in this chapter that describe different structures of source number estimation.

Terminologies

Definition 1 *A **centralized source number estimation scheme** is a processing structure that all the sensors send their data directly to a central processing unit where source number estimation is performed.*

In the centralized scheme, the information transmitted through the network is *raw data* collected by the sensors. The structure of centralized scheme is shown in Fig. 3.1(a).

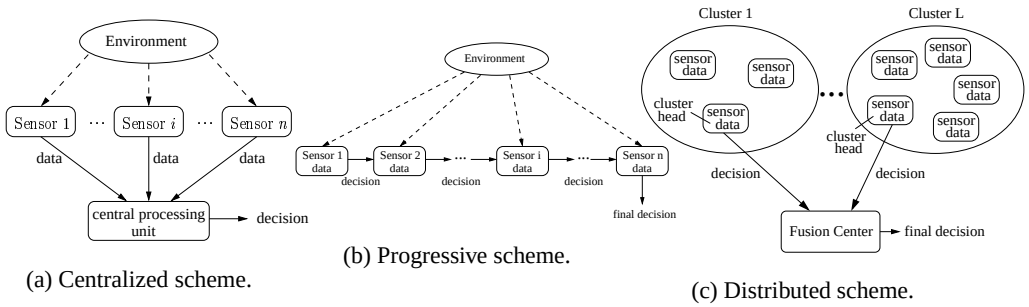


Figure 3.1: The structure of centralized scheme, progressive scheme and distributed scheme.

Definition 2 *A **progressive source number estimation scheme** is a processing structure that a group of sensors update the source number estimation result sequentially based on each sensor's local observation and the partial estimation result from its previous sensors in the sequence.*

In this scheme, the information transmitted through the network is the *estimation result* or *partial decision*. The structure of progressive scheme is shown in Fig. 3.1(b), where sensor i generates its partial decision based on its own observation and the partial decision received from sensor $i-1$.

Definition 3 *A **distributed** or **cluster-based source number estimation scheme** is a structure including two levels of processing: source number estimation within each cluster and decision fusion between different clusters.*

In this scheme, sensor nodes are divided into clusters with each sensor belonging to one and only one cluster, as illustrated in Fig. 3.1(c). Source number estimation is performed within each cluster in either centralized or progressive manner. The estimation results from different clusters are then sent to a fusion center to generate a final decision.

Problem Definition

Suppose there are m targets in the field generating independent source signals $s_i^{(t)}, i = 1, \ldots, m$ and n sensors recording signals $x_j^{(t)}, j = 1, \ldots, n$, where $t = 1, \ldots, T$ indicates the time index of the discrete-time signals. Then the sources and the observed mixtures at time t can be denoted as vectors $\mathbf{s}^{(t)} = [s_1^{(t)}, \ldots, s_m^{(t)}]^T$ and $\mathbf{x}^{(t)} = [x_1^{(t)}, \ldots, x_n^{(t)}]^T$, respectively. If we assume the mixing process is linear, the observations can then be represented as

$$\mathbf{x}^{(t)} = \mathbf{A}\mathbf{s}^{(t)} \tag{3.2}$$

and the sources as

$$\hat{\mathbf{s}}^{(t)} = \mathbf{W}\mathbf{x}^{(t)} \tag{3.3}$$

where $\mathbf{A}_{n \times m}$ is the unknown non-singular scalar mixing matrix and the unmixing matrix $\mathbf{W}$ is calculated as the Moore-Penrose pseudo-inverse,

$$\mathbf{W} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \tag{3.4}$$

Based on this linear, instantaneous mixing model, the source number estimation can be considered as a hypothesis testing problem, where $\mathcal{H}_m$ denotes the hypothesis on a possible number of targets and the goal is to find $\hat{m}$ whose corresponding hypothesis $\mathcal{H}_{\hat{m}}$ maximizes the posterior probability given only the observation $\mathbf{x}^{(t)}$,

$$\hat{m} = \arg\max_m P(\mathcal{H}_m|\mathbf{x}^{(t)}). \tag{3.5}$$

According to Bayes' theorem, the posterior probability of hypotheses can be written as

$$P(\mathcal{H}_m|\mathbf{x}^{(t)}) = \frac{p(\mathbf{x}^{(t)}|\mathcal{H}_m)P(\mathcal{H}_m)}{\sum_{all\ \mathcal{H}} p(\mathbf{x}^{(t)}|\mathcal{H})P(\mathcal{H})} \tag{3.6}$$

Assume the hypothesis $\mathcal{H}_m$ has a uniform distribution, then the measurement of posterior probability can be simplified to the calculation of likelihood $p(\mathbf{x}^{(t)}|\mathcal{H}_m)$. In other words, the objective can be reformatted as finding $\hat{m}$ that maximizes the likelihood function (the objective function).

Several other assumptions are also adopted for the linear mixing model [79]:

- At most one source is normally distributed.
- No sensor noise or only low additive noise signals are permitted.

3.2.2 Related Work

Independent Component Analysis (ICA)

Independent component analysis (ICA) algorithm is the most popular technique developed to solve the BSS problem. The goal of BSS and ICA is to extract independent source signals from their linear or nonlinear mixtures using a minimum of a priori information. ICA not only decorrelates the signals which is of second order statistics, but also reduces higher-order statistical dependencies, trying to make the signals as independent as possible.

There are two different research communities that have considered the analysis of independent components [79]. On one hand, the study of separating mixed sources observed in an array of sensors has been a classical and difficult signal processing problem. Herault and Jutten were first ones working on BSS who, in their seminal work on this topic, introduced an adaptive algorithm in a simple feedback architecture that was able to separate several unknown independent sources [58]. Their approach has been further developed by many other researchers. Comon was one of them who elaborated the concept of ICA and proposed cost functions related to the approximate minimization of mutual information between the sensors [32]. In parallel to BSS studies, unsupervised learning rules based on information theory were proposed whose goal was to maximize the mutual information between the inputs and outputs of a neural network [81]. Roth and Baram [117] and Bell and Sejnowski [13] independently

derived stochastic gradient learning rules for this maximization and applied them, respectively, to forecasting, time series analysis, and the blind separation of sources. In their work, Bell and Sejnowski put the BSS problem into an information-theoretic framework and demonstrated the separation and deconvolution of mixed sources [13].

Intuitively speaking, the key to estimating the ICA model is nongaussianity. Therefore, the definition and estimation of a contrast function which measures the nongaussianity of the independent components is necessary for the identifiability of the model. There are many different representations of the contrast function, where high-order cumulants, mutual information, and negentropy are the most important ones.

The fourth-order cumulant or kurtosis is the classical measure of nongaussianity of signals, defined for a zero-mean random variable u as $kurt(u) = E\{u^4\} - 3(E\{u^2\})^2$ [62, 63]. For a Gaussian random variable, kurtosis is zero; for densities peaked at zero, it is positive, and for flatter densities, negative. In [62], the authors derived an objective function in order to maximize or minimize kurtosis $kurt(\mathbf{W}^T\mathbf{x}^{(t)})$ where $\mathbf{W}$ is the corresponding weight matrix. However, kurtosis also has some drawbacks in practice, such as being very sensitive to outliers [63], which makes it not a robust measure of nongaussianity.

Mutual information, inspired by information theory, is a natural measure of the dependence between random variables, which makes it lead to the same principle of finding nongaussianity of the signals. The mutual information I between m random variables $y_i, i = 1, \ldots, m$ is defined as $I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y})$ where $H(\mathbf{y})$ is the differential entropy of the random variable. The minimization of mutual information corresponds to finding the most independent components which is the objective of the BSS problem. The Infomax principle [13] is derived from the minimization of mutual information (correspondingly, maximizing the output entropy) of a neural network with non-linear outputs.

Another very important measure of nongaussianity is given by negentropy which is based on the information theoretic quantity of differential entropy [63]. The negentropy of a random

vector $\mathbf{y}$ is defined as $J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$ where $H(\mathbf{y}$ is the differential entropy of the random vector $\mathbf{y}$ and $H(\mathbf{y}_{gauss})$ denotes the differential entropy of a Gaussian random variable $\mathbf{y}_{gauss}$ which is of the same covariance matrix as $\mathbf{y}$. The advantage of using negentropy as a measure of nongaussianity is that it is well justified by statistical theory. The FastICA algorithm [62] is developed to maximize the contrast function measured by negentropy. It uses a fixed point iteration scheme for finding a direction such that the projection of weight vectors maximizes an approximation of negentropy.

In the derivation of most ICA algorithms, it is assumed that the number of sources equals the number of observations so that the mixing/unmixing matrix is square and easy to be estimated. However, in a sensor network with sheer amount of sensor nodes densely deployed, this assumption is not always true. Hence, an important problem arises as to estimate the number of sources dynamically.

Source Number Estimation

Several approaches have been introduced to solve the source number estimation problem so far, some heuristic, others based on more principled approaches [69, 113, 115]. As discussed recently in [114], it has become clear that techniques of the latter category are superior, and heuristic methods may be seen at best as approximations to more detailed underlying principles. In this section, we focus on the discussion of principled source number estimation algorithms, which construct multiple hypotheses corresponding to different number of sources.

Bayesian Source Number Estimation: The Bayesian estimation method [115] uses a set of Laplace approximations to estimate the marginal integrals when calculating the posterior probability of source number hypotheses. This approach has a solid theoretical background and the objective function is easy to calculate, hence, it provides a practical solution for the source number estimation problem. The derivation of the classic centralized Bayesian approach is presented in Sec. 3.2.3.

Sample-based Source Number Estimation: Other than the Laplace approximation method, the posterior probabilities of specific hypotheses can also be evaluated using a sample-based approach. In this approach, a reversible-jump Markov chain Monte Carlo (RJ-MCMC) method is proposed to estimate the joint density over the mixing matrix $\mathbf{A}$, the hypothesized number of sources m, and the noise component R_n, which is denoted as $P(\mathbf{A}, m, R_n)$ [114, 113]. The basic idea is to construct a Markov chain which generates samples from the hypothesis probability and to use the Monte Carlo method to estimate the posterior probability from the samples. An introduction of Monte Carlo methods can be found in [84].

RJ-MCMC is actually a random-sweep Metropolis-Hastings method, where the transition probability of the Markov chain from state $(\mathbf{A}, m, R_n)$ to state $(\mathbf{A}', m', R_n')$ is

$$p = min\{1, \frac{P(\mathbf{A}', m', R_n'|\mathbf{x}^{(t)})}{P(\mathbf{A}, m, R_n|\mathbf{x}^{(t)})} \frac{q(\mathbf{A}, m, R_n|\mathbf{x}^{(t)})}{q(\mathbf{A}', m', R_n'|\mathbf{x}^{(t)})} J\}, \tag{3.7}$$

where $P(\cdot)$ is the posterior probability of the unknown parameters of interest, $q(\cdot)$ is a proposal density for moving from state $(\mathbf{A}, m, R_n)$ to state $(\mathbf{A}', m', R_n')$, and J is the ratio of Jacobians for the proposal transition between the two states [114]. More detailed derivation of this method is provided in [113].

Variational Learning: In recent years, the Bayesian inference problem shown in Eq. 3.6 is also tackled using another approximative method known as variational learning [8, 16]. In ICA problems, variables are divided into two classes: the visible variables v and the hidden variables h. An example of visible variables is the observation matrix $\mathbf{x}$; examples of hidden variables include an ensemble of the parameters of $\mathbf{A}$, the noise covariance matrix, any parameters in the source density models, and all associated hyperparameters such as the number of sources m [114]. Suppose $q(h)$ denotes the variational approximation to the posterior probability of the hidden variables $P(h|v)$, then the negative variational free energy, F, is defined as

$$F = \int q(h) \ln P(h|v) dh + H[q(h)] \tag{3.8}$$

where $H[q(h)]$ is the differential entropy of $q(h)$. The negative free energy F forms a strict lower bound on the evidence of the model, $\ln p(v) = \ln(\int p(v|h)p(h)dh)$. The difference between this variational bound and the true evidence is the Kullback-Leibler (KL) divergence between $q(h)$ and the true posteriors $P(h|v)$ [29]. Therefore, maximizing F is equivalent to minimizing the KL divergence, and this process provides a direct method for source number estimation.

Another promising source number estimation approach using variational learning is the so-called Automatic Relevance Determination (ARD) scheme [29]. The basic idea of ARD is to suppress sources that are unsupported by the data. For example, assume each hypothesized source has a Gaussian prior with separate variances, those sources that do not contribute to modeling the observations tend to have very small variances and the corresponding source models do not move significantly from their priors [114]. After eliminating those unsupported sources, the sustained sources give the true number of sources of interest.

Even though variational learning is a particularly powerful approximative approach, it is yet to be developed into a more mature form. In addition, it presents difficulties to estimate the true number of sources with noisy data.

3.2.3 Classic Centralized Bayesian Estimation Scheme

In this section, we briefly review the classic Bayesian source number estimation algorithm. As discussed in [115], maximizing the informativeness of the set of estimated sources may be achieved by making $\mathbf{W}$ as large as possible which requires some form of constraint. An alternative approach is to linearly map the observations $\mathbf{x}^{(t)}$ to a set of latent variables, $\mathbf{a}^{(t)}$, of the form $\mathbf{a}^{(t)} = \mathbf{W}\mathbf{x}^{(t)}$, followed by a non-linear transform from this latent space to the set of source estimations, $\hat{\mathbf{s}}^{(t)} = \phi(\mathbf{a}^{(t)})$. Generally speaking, the choice of the non-linear transform

is not critical and can be defined as:

$$\phi(\mathbf{a}^{(t)}) = -\tanh(\alpha \mathbf{a}^{(t)}) \tag{3.9}$$

where α is a scaling factor. Using the latent variable representation, the likelihood function of $\mathbf{x}$ conditioned on the mixing matrix $\mathbf{A}$, the choice of non-linearity, ϕ, and the noise R_n is

$$p(\mathbf{x}|\mathbf{A}, \phi, R_n) = \prod_t p(\mathbf{x}^{(t)}|\mathbf{A}, \phi, R_n) \tag{3.10}$$

By marginalizing over the latent variable space, the likelihood can be written as

$$p(\mathbf{x}^{(t)}|\mathbf{A}, \phi, R_n) = \int p(\mathbf{x}^{(t)}|\mathbf{A}, \phi, R_n, \mathbf{a}^{(t)}) \pi(\mathbf{a}^{(t)}) d\mathbf{a}^{(t)} \tag{3.11}$$

where $\pi(\cdot)$ is the assumed marginal distribution of the latent variables. For the choice $\phi(\mathbf{a}^{(t)}) = -\tanh(\alpha \mathbf{a}^{(t)})$, it is of the form

$$\pi(\mathbf{a}^{(t)}) = \frac{1}{Z(\alpha)} (\cosh(\alpha \mathbf{a}^{(t)}))^{\frac{1}{\alpha}} \tag{3.12}$$

$$\log(Z(\alpha)) = a \log(\frac{c}{\alpha} + 1) + b \tag{3.13}$$

where $a = 0.522,\ b = 0.692,\ c = 1.397.$

Assume the noise on each component has the same variance, given by $\frac{1}{\beta}$, then we may write

$$p(\mathbf{x}^{(t)}|\mathbf{A}, \phi, \beta, \mathbf{a}^{(t)}) = \frac{1}{H} \exp -\frac{\beta}{2} (\mathbf{x}^{(t)} - \mathbf{A}\mathbf{a}^{(t)})^2 \tag{3.14}$$

where

$$\frac{1}{H} = (\frac{\beta}{2\pi})^{\frac{n}{2}} \tag{3.15}$$

Further assume the integral in Eq. 3.11 is dominated by a sharp peak at $\mathbf{a}^{(t)} = \hat{\mathbf{a}}^{(t)}$, then a Laplace approximation of the marginal integral can be made and the log-likelihood evidence is

estimated as:

$$\log p(\mathbf{x}^{(t)}|\mathbf{A},\phi,\beta) = \log \pi(\hat{\mathbf{a}}^{(t)}) + \frac{1}{2}(n-m)\log(\frac{\beta}{2\pi}) - \frac{1}{2}\log|\mathbf{A}^T\mathbf{A}| - \frac{\beta}{2}(\mathbf{x}^{(t)} - \mathbf{A}\hat{\mathbf{a}}^{(t)})^2 \tag{3.16}$$

where $\hat{\mathbf{a}}^{(t)} = \mathbf{W}\mathbf{x}^{(t)}$. By integrating the likelihood function in Eq. 3.16 over the density function of the model parameter $\mathbf{A}$ and evaluating the integral using the Laplace approximation once more under the assumption that it is dominated by a sharp peak at $\mathbf{A} = \hat{\mathbf{A}}$, an estimate of the predictive density can be achieved:

$$p(\mathbf{x}^{(t)}|\phi,\beta) = \pi(\hat{\mathbf{a}}^{(t)})(\frac{\beta}{2\pi})^{\frac{1}{2}(n-m)}(\frac{\beta}{2\pi})^{-\frac{mn}{2}}\frac{\pi(\hat{\mathbf{A}})}{(\prod_{j=1}^{m}(\hat{a}_j^2)^n)^{\frac{1}{2}}}\exp\{-\frac{\beta}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2\} \tag{3.17}$$

The final stage of the analysis involves performing the marginal integral over the variance β. By using the maximum-likelihood estimation method, an estimate $\hat{\beta}$ can be optimized and the posterior distribution over β can be assumed to be sharply peaked around $\hat{\beta}$:

$$\hat{\beta} = \frac{1}{n-m}\langle(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2\rangle \tag{3.18}$$

Finally, taking logarithms of Eq. 3.17 with $\beta = \hat{\beta}$ and assuming the prior of $\mathbf{A}$ has the form,

$$P(\mathbf{A}) = \gamma^{-mn} \tag{3.19}$$

$$\gamma = 2\|\hat{\mathbf{A}}\|_{infty} \tag{3.20}$$

a data penalized log-likelihood function which is proportional to the posterior probability, can be written as:

$$\begin{aligned} L(m) &= \log p(\mathbf{x}^{(t)}|\mathcal{H}_m) \\ &= \log \pi(\hat{\mathbf{a}}^{(t)}) + \frac{n-m}{2}\log(\frac{\hat{\beta}}{2\pi}) - \frac{1}{2}\log|\hat{\mathbf{A}}^T\hat{\mathbf{A}}| - \frac{\hat{\beta}}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2 \end{aligned}$$

$$-\frac{mn}{2}\log(\frac{\hat{\beta}}{2\pi}) - \frac{n}{2}(\sum_{j=1}^{m}\log\hat{a}_j^2) - mn\log\gamma \tag{3.21}$$

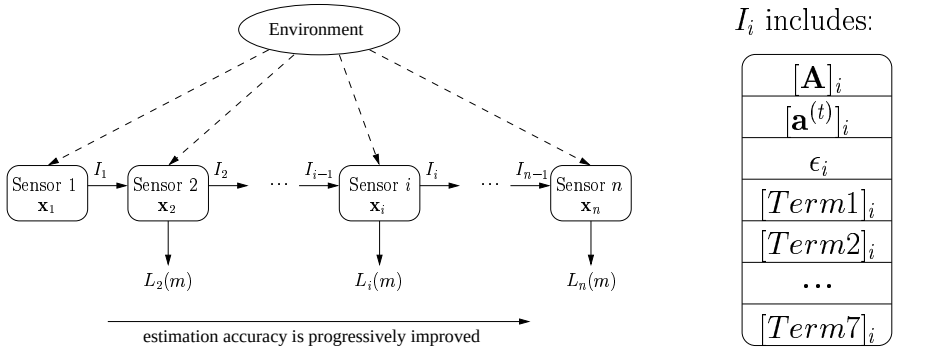
As you can see from Eq. 3.21 that in order to calculate the log-likelihood $L(m)$, the raw data $\mathbf{x}^{(t)}$ from all participating sensor nodes are needed. The transmission of raw data consumes energy and bandwidth which is in contradictory to the requirement of sensor networks. Therefore, the most challenging problem of source number estimation in sensor networks is to be able to estimate the log-likelihood $L(m)$ locally without direct usage of other sensor observations. This is the motivation of the progressive estimation approach presented in Sec. 3.2.4.

3.2.4 Progressive Bayesian Estimation Approach

In order to accommodate the unique energy efficiency requirement of sensor networks in performing target number estimation, a progressive approach is derived to evaluate the objective function of Eq. 3.21 based on the iterative relationship between sensors, i.e. each sensor i updates the log-likelihood evaluation only based on its local observation $x_i^{(t)}$ and the information I_{i-1} transmitted from its previous sensor $(i-1)$ as shown in Fig. 3.2(a).

To carry out the updating process successfully, two problems must be solved: 1) How to update the mixing matrix $\mathbf{A}$ iteratively and keep the independence of source signals; and 2) How to decompose the log-likelihood estimation into different components that depend on the previous log-likelihood estimation and the local observation, respectively.

In the progressive scheme shown in Fig. 3.2(a), upon receiving a partial estimation result I_{i-1} from its previous sensor $(i-1)$, sensor i will update the log-likelihood $L_i(m)$ and the mixing matrix $\mathbf{A}$ corresponding to different source number hypotheses and then transmit the updated results I_i to its next sensor $(i+1)$. The updating rules at sensor i is denoted as $\mathcal{D}_i(x_i^{(t)}; I_{i-1})$, where $x_i^{(t)}$ is the observation of sensor i at time t and I_{i-1} denotes the information received from sensor $(i-1)$. As shown in Fig. 3.2(b), the information transmitted by sensor i, I_i, includes the updated mixing matrix $[\mathbf{A}]_i$, the estimated latent variable $[\mathbf{a}^{(t)}]_i$, the



(a) Structure. (b) Transmitted information.

Figure 3.2: Progressive source number estimation scheme.

accumulated estimation error ϵ_i (defined in Sec. 3.2.4), and all seven terms in Eq. 3.21 updated at sensor i, $[Term1]_i, [Term2]_i, \cdots, [Term7]_i$.

Progressive Estimation of the Mixing Matrix A

At the first sensor, the mixing matrix $\mathbf{A}$ is initialized randomly. During the progressive implementation, sensor $(i-1)$ modifies matrix $\mathbf{A}_{(i-1)\times m}$ locally, and sends the resulting matrix to sensor i. After sensor i receives the information, it first adds one more dimension (an extra row) to $\mathbf{A}$ with random numbers, and then finds the optimal estimate of $\mathbf{A}_{i\times m}$ using the BFGS optimization method which is one special type of quasi-Newton methods [123, 106].

Progressive Estimation of the Accumulated Error

We refer to the component $\sum_{j=1}^{i-1}(x_j^{(t)} - \sum_{k=1}^{m} A_{j,k}\hat{a}_k^{(t)})^2$ as the *accumulated estimation error* ϵ_{i-1} at sensor $(i-1)$ which is the squared difference between the sensor observation and the mixture of estimated source signals. Basically, it originates from the estimation error of the mixing matrix $\mathbf{A}$. This item needs to be carried from sensor to sensor and its updating rule at

sensor i is

$$\epsilon_i = \sum_{j=1}^{i}(x_j^{(t)} - \sum_{k=1}^{m} A_{j,k}\hat{a}_k^{(t)})^2 = \epsilon_{i-1} + (x_i^{(t)} - \sum_{k=1}^{m} A_{i,k}\hat{a}_k^{(t)})^2 \tag{3.22}$$

Progressive Estimation of the Log-likelihood Function

In order to estimate the log-likelihood function in Eq. 3.21 progressively, we employ the iterative relationship between sensors, i.e., the estimation at sensor i, $L_i(m)$, is a function of the estimation at sensor $(i-1)$, $L_{i-1}(m)$, and the local observation $x_i^{(t)}$. The updating rule can be formulated as

$$L_i(m) = L_{i-1}(m) + f(x_i^{(t)}) \tag{3.23}$$

In this section, we derive the updating function f by evaluating the seven terms of the log-likelihood function separately.

Term 1: The first term in Eq. 3.21, $\log \pi(\hat{a}_k^{(t)})$, accounts for the marginal distribution of the latent variables. In normal cases, the distribution of one latent variable $\hat{a}_k$ can be assumed to have the form [115]

$$\pi(\hat{a}_k^{(t)}) = \frac{1}{Z(\alpha)[\cosh(\alpha\hat{a}_k^{(t)})]^{\frac{1}{\alpha}}} \tag{3.24}$$

where $\log Z(\alpha) = a\log(\frac{c}{\alpha}+1)+b$, α is a scaling factor, $a = 0.522$, $b = 0.692$, and $c = 1.397$. Suppose the mixing matrix at sensor i is $\mathbf{A}^i$, then $\mathbf{W}^i = [(\mathbf{A}^i)^T\mathbf{A}^i]^{-1}(\mathbf{A}^i)^T$, and $\mathbf{w}_k^i$ is the kth row of $\mathbf{W}^i$. Hence, at sensor $(i-1)$, the latent variable $\hat{a}_k^{(t)}$ is of the form

$$\hat{a}_k^{(t)} = \sum_{j=1}^{i-1} w_{k,j}x_j^{(t)} \tag{3.25}$$

and the first term in Eq. 3.21 can be calculated as:

$$\begin{aligned}
[Term1]_{i-1} &= [\log \pi(\hat{a}_k^{(t)})]_{i-1} = -\log Z(\alpha) - \frac{1}{\alpha}\log[\cosh(\alpha\hat{a}_k^{(t)})] \\
&= -\log Z(\alpha) - \frac{1}{\alpha}\log[\frac{\exp(\alpha\hat{a}_k^{(t)}) + \exp(-\alpha\hat{a}_k^{(t)})}{2}]
\end{aligned}$$

$$= \quad -\log Z(\alpha) - \frac{1}{\alpha}\log[\exp(\alpha\hat{a}_k^{(t)})(1+\exp(-2\alpha\hat{a}_k^{(t)})] + \frac{1}{\alpha}\log 2 \quad (3.26)$$

We can restrict the initial choice of $\mathbf{A}^{i-1}$ such that $|\exp(-2\alpha\hat{a}_k^{(t)})| < 1$. According to the Taylor expansion,

$$\log[1+\exp(-2\alpha\hat{a}_k^{(t)})] = \sum_{p=1}^{\infty}\frac{(-1)^{p-1}}{p}[\exp(-2\alpha\hat{a}_k^{(t)})]^p \quad (3.27)$$

Then Eq. 3.26 becomes:

$$[Term1]_{i-1} = -\log Z(\alpha) - \hat{a}_k^{(t)} - \frac{1}{\alpha}\sum_{p=1}^{\infty}\frac{(-1)^{p-1}}{p}[\exp(-2\alpha\hat{a}_k^{(t)})]^p + \frac{1}{\alpha}\log 2 \quad (3.28)$$

If only consider the calculation to the first order precision, i.e., $p = 1$, and substitute Eq. 3.25 into Eq. 3.28, Term 1 calculated at sensor $(i-1)$ is

$$[Term1]_{i-1} \approx -\log Z(\alpha) - \sum_{j=1}^{i-1} w_{k,j}x_j^{(t)} - \frac{1}{\alpha}\exp(-2\alpha\sum_{j=1}^{i-1} w_{k,j}x_j^{(t)}) + \frac{1}{\alpha}\log 2 \quad (3.29)$$

Similarly, Term 1 calculated at sensor i can be written as:

$$\begin{aligned}[Term1]_i &= [\log\pi(\hat{a}_k^{(t)})]_i = -\log Z(\alpha) - \frac{1}{\alpha}\log(\cosh(\alpha\sum_{j=1}^{i} w_{k,j}x_j^{(t)})) \\ &\approx -\log Z(\alpha) - \sum_{j=1}^{i} w_{k,j}x_j^{(t)} - \frac{1}{\alpha}\exp(-2\alpha\sum_{j=1}^{i} w_{k,j}x_j^{(t)}) + \frac{1}{\alpha}\log 2 \quad (3.30)\end{aligned}$$

Compare between Eq. 3.29 and Eq. 3.30, the updating rule for Term 1 at sensor i can be derived as

$$[Term1]_i = [Term1]_{i-1} - w_{k,i}x_i^{(t)} - \frac{1}{\alpha}\exp(-2\alpha[\hat{a}_k^{(t)}]_{i-1})[\exp(-2\alpha w_{k,i}x_i^{(t)}) - 1] \quad (3.31)$$

Term 2: The second term in Eq. 3.21, $\frac{n-m}{2}\log(\frac{\hat{\beta}}{2\pi})$, takes into account the noise variance

β, which is estimated by the squared errors between the real observations and their estimated counterparts. At sensor $(i-1)$, the noise variance is calculated as

$$\hat{\beta} = \frac{1}{i-1-m}\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2 \tag{3.32}$$

where A_j denotes the jth row of the mixing matrix $\mathbf{A}$ and m is the number of sources. Therefore, Term 2 at sensor $(i-1)$ is

$$\begin{aligned}
[Term2]_{i-1} &= [\frac{i-1-m}{2}\log(\frac{\hat{\beta}}{2\pi})]_{i-1} \\
&= -\frac{i-1-m}{2}\log(i-1-m) + \frac{i-1-m}{2}\log[\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2] \\
&\quad -\frac{i-1-m}{2}\log 2\pi
\end{aligned} \tag{3.33}$$

By using the Taylor expansion on component $\log[\sum_{j=1}^{i}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2]$ with the first order precision, the updating rule of Term 2 at sensor i is derived as:

$$\begin{aligned}
[Term2]_i &= -\frac{i-m}{2}\log(i-m) + \frac{i-m}{2}\log[\sum_{j=1}^{i}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2] - \frac{i-m}{2}\log 2\pi \\
&\approx -\frac{i-m}{2}\log(i-m) + \frac{i-m}{2}\log[\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2] \\
&\quad +\frac{i-m}{2}\cdot\frac{(x_i^{(t)} - A_i\hat{\mathbf{a}}^{(t)})^2}{\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2} - \frac{i-m}{2}\log 2\pi \\
&= \frac{i-m}{i-1-m}[Term2]_{i-1} + \frac{i-m}{2}\log(\frac{i-1-m}{i-m}) \\
&\quad +\frac{i-m}{2}\cdot\frac{(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2}{\epsilon_{i-1}}
\end{aligned} \tag{3.34}$$

Term 3: The third term in Eq. 3.21, $-\frac{1}{2}\log|\hat{\mathbf{A}}^T\hat{\mathbf{A}}|$, only depends on the updating rule of mixing matrix $\mathbf{A}$, which has been discussed in Sec. 3.2.4.

Term 4: The fourth term in Eq. 3.21 can be estimated at sensor $(i-1)$ as

$$[Term4]_{i-1} = [-\frac{\hat{\beta}}{2}(\mathbf{x}^{(t)} - \hat{\mathbf{A}}\hat{\mathbf{a}}^{(t)})^2]_{i-1} = -\frac{1}{2(i-1-m)}[\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2]^2 \quad (3.35)$$

Since the updating rule for Term 4 at sensor i involves the estimation of noise variance β, it is similar to that of Term 2, which is derived as:

$$\begin{aligned}
[Term4]_i &= -\frac{1}{2(i-m)}[\sum_{j=1}^{i}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2]^2 \\
&= -\frac{1}{2(i-m)}[\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2]^2 + 2(x_i^{(t)} - A_i\hat{\mathbf{a}}^{(t)})^2\sum_{j=1}^{i-1}(x_j^{(t)} - A_j\hat{\mathbf{a}}^{(t)})^2 \\
&\quad +(x_i^{(t)} - A_i\hat{\mathbf{a}}^{(t)})^4 \\
&= \frac{i-1-m}{i-m}[Term4]_{i-1} + 2\epsilon_{i-1}(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2 + (x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^4 \quad (3.36)
\end{aligned}$$

Term 5: The fifth term is also affected by the noise variance $\hat{\beta}$. Using the Taylor expansion formula as shown in Eq. 3.27, the updating rule of Term 5 at sensor i is:

$$\begin{aligned}
[Term5]_i &= [\frac{im}{2}\log(\frac{\hat{\beta}}{2\pi})]_i \\
&= \frac{i}{i-1}[Term5]_{i-1} + \frac{im}{2}\log(\frac{i-1-m}{i-m}) \\
&\quad +\frac{im}{2}\cdot\frac{(x_i^{(t)} - \sum_{k=1}^{m}A_{i,k}\hat{a}_k^{(t)})^2}{\epsilon_{i-1}} \quad (3.37)
\end{aligned}$$

Term 6: This term in Eq. 3.21 accounts for the estimated latent variables and its updating rule at sensor i employing the Taylor expansion with the 1st order precision in Eq. 3.27 can be written as:

$$[Term6]_i = [\frac{i}{2}\sum_{k=1}^{m}\log(\hat{a}_k^{(t)})^2]_i = \frac{i}{2}\sum_{k=1}^{m}\log(\sum_{j=1}^{i-1}w_{k,j}x_j^{(t)} + w_{k,i}x_i^{(t)})^2$$

$$\approx \frac{i}{2}\sum_{k=1}^{m}[\log(\sum_{j=1}^{i-1} w_{k,j}x_j^{(t)})^2 + \frac{(w_{k,i}x_i^{(t)})^2 + 2w_{k,i}x_i^{(t)}\sum_{j=1}^{i-1} w_{k,j}x_j^{(t)}}{(\sum_{j=1}^{i-1} w_{k,j}x_j^{(t)})^2}]$$

$$= \frac{i}{i-1}[Term6]_{i-1} + \frac{i}{2}\sum_{k=1}^{m}\frac{(w_{k,i}x_i^{(t)})^2 + 2w_{k,i}x_i^{(t)}[\hat{a}_k^{(t)}]_{i-1}}{([\hat{a}_k^{(t)}]_{i-1})^2} \quad (3.38)$$

Term 7: The last term in Eq. 3.21 is of the form $im\log\gamma$, where $\gamma = 2\|\hat{\mathbf{A}}\|_\infty$. This term only depends on the updating rule of matrix $\mathbf{A}$.

According to the update rules of the mixing matrix, the log-likelihood, and the estimation error at sensor i, the information to be transmitted from sensor i to sensor $i+1$ includes the estimated mixing matrix $[\hat{\mathbf{A}}]_i$, the estimated latent variables $[\hat{\mathbf{a}}^{(t)}]_i$, the estimation error ϵ_i, and each term in Eq. 3.21, i.e., $I_i = \{[\hat{\mathbf{A}}]_i, [\hat{\mathbf{a}}^{(t)}]_i, \epsilon_i, [Term1]_i, \cdots, [Term7]_i\}$. We denote this term as the *updating information* I hereafter. The progressive estimation algorithm is summarized in Algorithm 1 .

Implement the Progressive Approach Using Mobile Agent Framework

The progressive source number estimation algorithm exhibits energy efficiency over the classic centralized scheme in that it avoids the transmission of raw data from all the sensors to a central unit. In the progressive framework, each sensor processes its data locally and only sends the updating information I. However, direct implementation of this approach also presents some drawbacks: 1) Each sensor needs to keep a copy of the executable code (updating function) to update I locally. Upon receiving information from sensor $i-1$, sensor i needs a mechanism to call the updating function; 2) The usage of a pre-defined order of sensors cannot accommodate the dynamic environment in which sensor networks usually operate. In order to successfully implement the progressive approach, we present a mobile agent based framework that we have developed for collaborative target classification and information fusion in sensor networks [110].

In classical distributed sensor networks, it is assumed that all the local sensors communicate

Algorithm 1: Progressive source number estimation algorithm.

/*Initialization*/
At sensor $i = 1$, for each possible m:
Initialize $\mathbf{A}_{1\times m}$ using random numbers;
Compute $\mathbf{W}$ and $\mathbf{a}$;
Compute estimation error ϵ;
Compute each term in Eq. 3.21;
Compute $L(m)$;

/* Progressive Update */
$i = 2$;
while $\max L(m) < threshold$ **do**
    Send $\mathbf{A}$, latent variables $\mathbf{a}$, estimation error ϵ and the seven terms in Eq. 3.21 to sensor i;

    At sensor i, for each possible m:
    Add one row to $\mathbf{A}$ with random numbers;
    while *!converge* **do**
        Update $\mathbf{A}$ using BFGS method;
    end
    Update the accumulated estimation error ϵ;
    Update each term in Eq. 3.21;
    Compute $L(m)$;
    Increase i by 1;
end

/* Generate the final estimation */
Decide $m = \arg\max L(m)$;
Output m;

all their data to a central processor that performs data processing tasks. As discussed in Sec. 1.4, the client/server model has several drawbacks that have limited its usage in sensor networks, such as the requirements of large amount of network bandwidth, live network connection all the time, and precise consideration of the network traffic, the number of clients and servers, transaction volumes, etc. [108, 109]. On the contrary to the client/server model, we proposed an improved DSN architecture using mobile agent which is referred to as the mobile agent based DSN (MADSN) [108]. MADSN adopts the mobile agent computing paradigm that data stay at the local site, while the processing task (code) is moved to the data sites. By transmitting the computation engine instead of data, MADSN offers several important benefits [78, 107, 108, 134] such as reduced network bandwidth, better network scalability, extendibility, and stability.

In [148], the authors presented a modified mathematical model to perform analytical study of the data transfer time and simulated several scenarios using GloMoSim to compare the performance of the client/server-based paradigm and the mobile-agent-based paradigm. It was shown that the mobile-agent-based computing performed better in the context of sensor networks with large amount of sensors, unreliable communication links, and reduced bandwidth. It also increased the stability of the sensor network.

Generally speaking, a mobile agent is a program that can migrate from sensor to sensor performing information processing autonomously. The structure of a mobile agent is composed of four attributes: identification, itinerary, data, and processing code [109], as shown in Fig. 3.3. *Identification* uniquely identifies a mobile agent. *Itinerary* is the route of migration, i.e., the order of sensors the mobile agent visits. It can be fixed or dynamically determined. *Data* is the mobile agent's data buffer which carries the information transmitted from one sensor to another. In the progressive estimation, data buffer carries the updating information I. *Processing code* is the executables that perform information update when a mobile agent arrives at a local sensor. The processing code is carried by the mobile agent. A mobile agent daemon is executed at each sensor node listening to the network for incoming agents. When a valid mobile agent arrives,

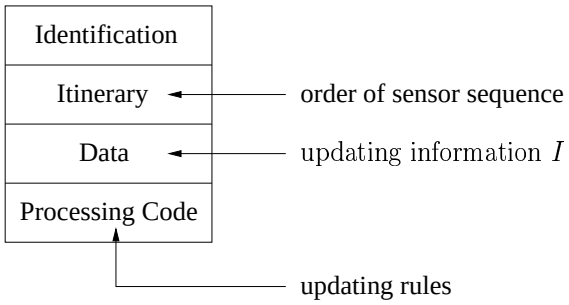


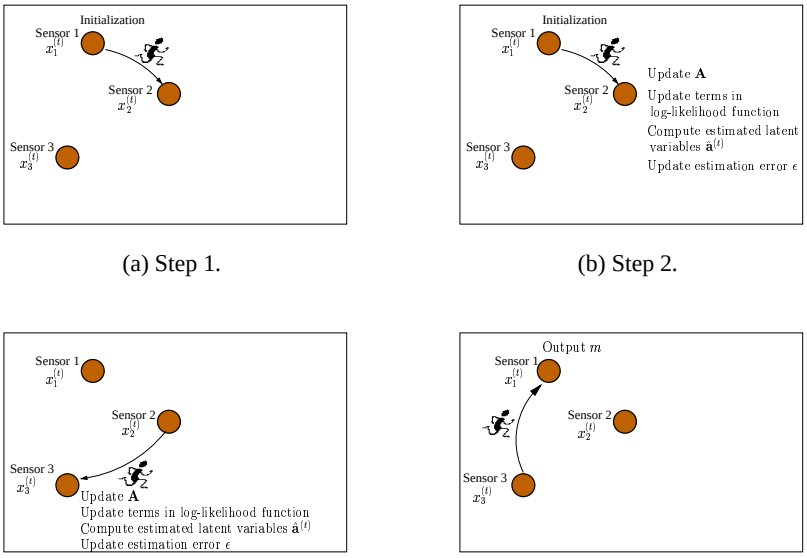
Figure 3.3: Structure of a mobile agent.

the daemon recovers the agent from the binary stream and executes the processing code.

Consider an example where the mobile agent migrates within a group of 3 sensors to implement the progressive estimation algorithm. The procedure of mobile agent based estimation is illustrated in Fig. 3.4. First, after initialization, sensor 1 dispatches a mobile agent to sensor 2, carrying the updating information I generated at sensor 1 as shown in Fig. 3.4(a). When the mobile agent arrives at sensor 2, it uses the local observation $x_2^{(t)}$ to update the estimation information, as shown in Fig. 3.4(b). Before sending out the mobile agent to the next sensor, the maximum of log-likelihoods corresponding to different source number hypotheses is evaluated and compared to a threshold. If it is beyond the threshold, which means the information available is sufficient to estimate the true number of sources, then the mobile agent will return to sensor 1 carrying the final result $\hat{m} = \arg\max L(m)$. Otherwise, the mobile agent continues its migration until the estimation accuracy is beyond the threshold or all sensors have been visited (as shown in Fig. 3.4(c) and (d)).

3.2.5 Distributed Source Number Estimation Scheme

Generally speaking, a sensor network consists of hundreds or thousands of low-cost sensor nodes that integrate multiple sensing modalities, data processing capability, and wireless communication. Each sensor node is able to independently sense the environment, but through



(a) Step 1.

(b) Step 2.

(c) Step 3.

(d) Step 4.

Figure 3.4: Procedures of mobile agent based progressive estimation.

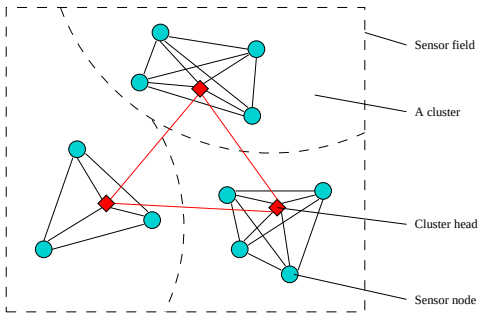


Figure 3.5: An example of clustered sensor network model.

collaboration they can achieve complex information gathering and dissemination tasks. Since sensor nodes are usually battery-powered and in most circumstances, they have to operate remotely in the field, it is difficult, if not impossible, to replace or recharge the battery in real time. Therefore, the most important issue in sensor network development is to conserve energy and prolong the lifetime of the entire network. A lot of research has been done on the energy consumption issue within each layer of sensor network protocol stack design, and it is shown that the wireless communication devices such as RF radios consume the most energy in sensor node operation. In a sensor network with a large amount of sensor nodes scattered in the field, two sensors can be far away from each other. For a multi-hop wireless network, the data traffic between two sensors far away needs to involve a large amount of intermediate "forwarding" sensors that consume extra energy. To tackle this problem, we develop a cluster-based distributed algorithm for multiple target detection by dividing the sensor nodes into different clusters.

Since sensor network clustering method is not the focus of our work, in the development of the distributed source number estimation scheme, we assume that a clustering protocol has been applied and the sensor nodes have organized themselves into clusters with each node assigned to one and only one cluster. Local nodes can communicate with other nodes within the same cluster, and different clusters communicate through a cluster head elected from each cluster. An example of a clustered sensor network is illustrated in Fig. 3.5.

The distributed scheme is accomplished in two levels: First, source number estimation is conducted within each cluster using either the centralized or progressive scheme. In the centralized scheme, each sensor within a cluster sends its observation to the cluster head where the log-likelihood of each source number hypothesis is estimated. While in the progressive scheme, the mobile agent framework is implemented. After a mobile agent is dispatched from the cluster head, it migrates among the sensor nodes in this cluster and updates the log-likelihood estimations of all possible hypotheses. Second, a fusion rule is applied to combine the posterior probability of each source number hypothesis estimated by each cluster. In this section, a fusion rule based on Bayes' theorem is derived to combine the results from different clusters.

Posterior Probability Fusion based on Bayes' Theorem

Consider the environment as a random variable $\mathbf{r}$, then the observation of sensor i at time instant t, $x_i^{(t)}$, is a sample of the random variable $\mathbf{r}$. If assume the observations from different sensors are conditionally independent, then

$$p(x_1^{(t)}, x_2^{(t)}, \cdots, x_n^{(t)}|\mathcal{H}) = \prod_{i=1}^{n} p(x_i^{(t)}|\mathcal{H}) \tag{3.39}$$

where $\mathcal{H}$ is a contional hypothesis and n is the number of sensors. Suppose sensor nodes are divided into L clusters, then the observations from cluster l, $l = 1, \cdots, L$, can be denoted as $\mathbf{x}_l^{(t)} = \{x_{l1}^{(t)}, \cdots, x_{ln}^{(t)}\}$ where $\{l1, \cdots, ln\}$ are the sensors belonging to cluster l. According to Eq. 3.39, the conditional probability given a hypothesis of source number is

$$p(\mathbf{x}^{(t)}|\mathcal{H}_m) = \prod_{l=1}^{L} p(\mathbf{x}_l^{(t)}|\mathcal{H}_m) \tag{3.40}$$

According to Bayes' theorem, the fused posterior probability can be written as

$$P(\mathcal{H}_m|\mathbf{x}^{(t)}) = \frac{p(\mathbf{x}^{(t)}|\mathcal{H}_m)P(\mathcal{H}_m)}{p(\mathbf{x}^{(t)})} = \frac{\prod_{l=1}^{L} p(\mathbf{x}_l^{(t)}|\mathcal{H}_m)P(\mathcal{H}_m)}{\prod_{l=1}^{L} p(\mathbf{x}_l^{(t)})} \tag{3.41}$$

If assume equal prior probability, then

$$P(\mathcal{H}_m|\mathbf{x}^{(t)}) \propto \frac{\prod_{l=1}^{L} p(\mathbf{x}_l^{(t)}|\mathcal{H}_m)}{\prod_{l=1}^{L} p(\mathbf{x}_l^{(t)})} \tag{3.42}$$

and

$$\log P(\mathcal{H}_m|\mathbf{x}^{(t)}) = \sum_{l=1}^{L} c_l \log p(\mathbf{x}_l^{(t)}|\mathcal{H}_m) \tag{3.43}$$

where c_l is the weight of cluster l in determining the log-likelihood. It reflects the physical characteristic of the clustering in the sensor network and is application-specific. For example, in the case of distributed multiple target detection using acoustic signals, the propagation of acoustic signals follows the energy decay model that the detected energy is inversely proportional to the square of the distance between the source and the sensor node, i.e., $E_{sensor} \propto \frac{1}{d^2} E_{source}$. Therefore, the weight c_l can be considered as the relative detection sensitivity of the sensor nodes in cluster l and is proportional to the average energy captured by the sensor nodes

$$c_l = \frac{1}{K_l} \sum_{k=1}^{K_l} E_k = \frac{1}{K_l} \sum_{k=1}^{K_l} \frac{1}{d_k^2} \tag{3.44}$$

where K_l denotes the number of sensor nodes in cluster l.

Dempster's Rule of Combination

In order to show the effectiveness of the Bayesian posterior probability fusion method, the Dempster's rule of combination is implemented as an alternative in the distributed estimation hierarchy. The Dempster's rule of combination [122] is a mature data fusion algorithm. It utilizes probability intervals and uncertainty intervals to determine the likelihood of hypotheses based on multiple evidence. It can also assign measures of belief to combinations of hypotheses. The Dempster's combination rule is developed based on a set of mutually exclusive alternatives, U [122]. For each subset Y of U, there are three parameters need to be specified: a basic prob-

ability, a belief, and a plausible belief. In the developed distributed source number estimation algorithm, the Dempster's rule is implemented by setting the posterior probability estimated in each cluster as one of the basic probability assignment and by combining them into a global basic probability assignment:

$$P(\mathcal{H}_m|\mathbf{x}^{(t)}) = \frac{\sum_{all\ \mathcal{H}_f \cap \mathcal{H}_g = \mathcal{H}_m, i \neq j} P(\mathcal{H}_f|\mathbf{x}_i^{(t)}) * P(\mathcal{H}_g|\mathbf{x}_j^{(t)})}{1 - \sum_{all\ \mathcal{H}_f \cap \mathcal{H}_g = \emptyset, i \neq j} P(\mathcal{H}_f|\mathbf{x}_i^{(t)}) * P(\mathcal{H}_g|\mathbf{x}_j^{(t)})} \tag{3.45}$$

After distributed source number estimation, the mixing matrix and the independent components can be derived from the mixture signals. Each estimated source signal can be considered as corresponding to a specific target. Therefore, classification can be performed on each estimated source and a target type prediction will be given.

Chapter 4

Collaborative Target Classification Hierarchy in Sensor Networks

Sensor networks usually consist of thousands of sensor nodes that are densely deployed in the field. When an event is detected, the captured signals from multiple sensor nodes need to be processed in order to classify the corresponding targets and the results from different sensors need to be fused and sent to the operator. The signal processing and fusion algorithms in sensor networks need to meet two requirements: First, they should be robust and fault tolerant so that they can handle uncertainty and node failure effectively [73]. The redundancy in sensor readouts are normally used to provide error tolerance. Secondly, the algorithm also need to meet the energy efficiency requirement. This requirement, however, is in conflict with the fault tolerance requirement since redundancy is to be reduced as much as possible in order to achieve energy efficiency.

In this chapter, we present a general purpose multi-modality, multi-sensor fusion hierarchy to perform collaborative target classification. Generally speaking, each sensor node is an integrated entity of a signal processing element and multiple sensing modalities. Each modality is able to sense a different aspect of its surroundings. For example, microphones capture the

acoustic waves that are generated by the rotation of different parts of ground vehicles, and geophones capture the seismic waves resulted from the vibration of different components within vehicles coupled to the ground. The developed signal and information fusion hierarchy includes four levels of processing: local signal processing on data collected from a single sensing modality, temporal fusion of local signal processing results over time, multi-modality fusion of results from different sensing modalities, and multi-sensor fusion of information from different sensor nodes. During local processing, features are extracted from the signals of each modality on each sensor node and a decision on the target class is made using pattern recognition algorithms. Then the local classification results of each modality are fused for a duration of time. The third level of the hierarchy is the multi-modality fusion on all the local decisions drawn from different sensing modalities within each sensor node, and finally, the decisions from all the sensor nodes are integrated on top of a mobile agent framework (MAF).

4.1 Problem Formulation

Assume $\mathbf{x}_i^j = [x_i^j(t_1), \ldots, x_i^j(t_k), \ldots, x_i^j(t_T)]$ denotes the sensor readout of a certain event from the i-th sensor node, j-th sensing modality over a time period from t_1 to t_T with t_k ($k = 1, \ldots, T$) representing a sub-event *interval*. Without loss of generality, it is assumed that there are N sensor nodes and each sensor has M sensing modalities on board. The observed sensor readouts are processed in a hierarchical manner, which includes: local processing, temporal fusion, multi-modality fusion, and multi-sensor fusion. The hierarchy is illustrated in Fig. 4.1.

Local signal processing is at the lowest level of the hierarchy. It occurs at a single sensor node and operates on data collected from individual sensing modality. Local processing is initiated by event detection. Since each event usually covers a period of time and the captured signal in this time period might be non-stationary, the data corresponding to an event (e.g. 10 seconds of data) is usually divided into sub-intervals $x_i^j(t_k)$ on which local processing is

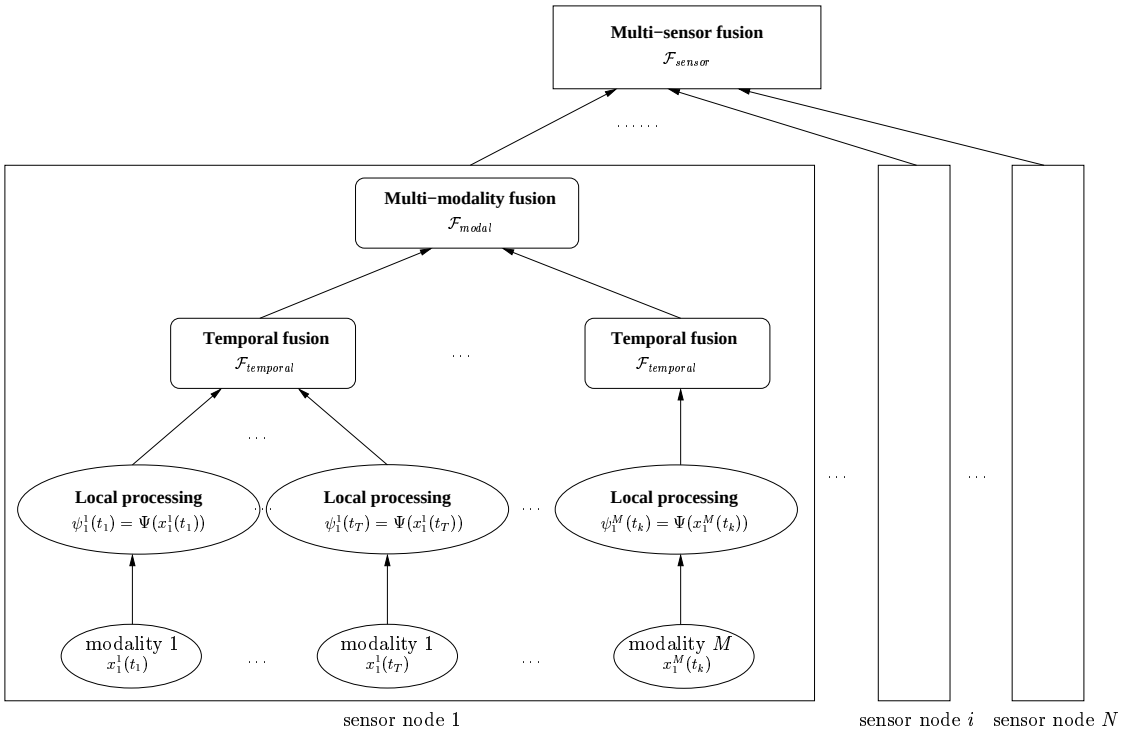


Figure 4.1: The hierarchy of the signal and information processing algorithm.

applied,

$$\psi_i^j(t_k) = \Psi(x_i^j(t_k)) \tag{4.1}$$

where $\Psi(\cdot)$ denotes the local processing algorithm which is task-adaptive.

Temporal fusion is used to fuse those local processing results of sub-intervals corresponding to an event over time using function $\mathcal{F}_{temporal}$:

$$\overline{\psi_i^j} = \mathcal{F}_{temporal}(\psi_i^j(t_1), \cdots, \psi_i^j(t_k), \cdots, \psi_i^j(t_T)) \tag{4.2}$$

where $\overline{\psi_i^j}$ is the interpretation of the event provided by the jth sensing modality of the ith sensor node.

Normally there are multiple sensing modalities equipped on each sensor node in order to compensate for each other's sensing limitation to provide an overall measurement of the environment. Therefore, the results from the appropriate sensing modalities are further integrated using function $\mathcal{F}_{modal}$:

$$\xi_i = \mathcal{F}_{modal}(\overline{\psi_i^1}, \cdots, \overline{\psi_i^j}, \cdots, \overline{\psi_i^M}) \tag{4.3}$$

These multi-modality fusion results then undergo the highest-level of fusion across different sensor nodes, multi-sensor fusion, and generate the final result:

$$\eta = \mathcal{F}_{sensor}(\xi_1, \cdots, \xi_i, \cdots, \xi_N) \tag{4.4}$$

The enabling algorithms for each level of the hierarchy are discussed in the following sections. The local processing algorithm for collaborative target classification in ground sensor networks is presented first since local processing is application oriented, then the information fusion hierarchy is described in a upward manner.

4.2 Local Signal Processing

The local target classification algorithm operates on the raw data $x_i^j(t_k)$ sensed from the i-th sensor node, j-th sensing modality over a sub-event interval t_k. In general, some *a-priori* knowledge of the statistical characteristics of different target signatures is required. In our experiment, each sensor node is equipped with two sensing modalities, microphone for the acoustic signal and geophone for the seismic signal. Both the acoustic and seismic signals emitted by ground vehicles are strongly non-stationary because of the interference from many factors, such as the speed of the target, noise from various moving parts and frictions, and environmental effects, etc. [109, 141]. Therefore, it is crucial to extract representative and robust features in order to classify targets correctly.

4.2.1 Feature Extraction

Feature extraction is the process to obtain signal characteristics from the time series data. It can be considered as a data compression process which removes irrelative information and preserve relevant one from the raw data [138]. Feature extraction plays an important role in target classification problem since the performance of the classifier largely depends on the quality of the feature vectors. In order to conquer the non-stationarity of the captured signals, we use features derived from both the frequency and time-frequency domains. The block diagram of the feature extraction procedure is shown in Fig. 4.2.

Each feature vector includes 26 elements among which 14 are derived from the power spectral density (PSD) and the others from the wavelet coefficients of the time series data. Then normalization is performed on the feature vectors to eliminate the scaling effects among different elements, and the dimension of feature vectors are reduced using the principal component analysis (PCA) algorithm.

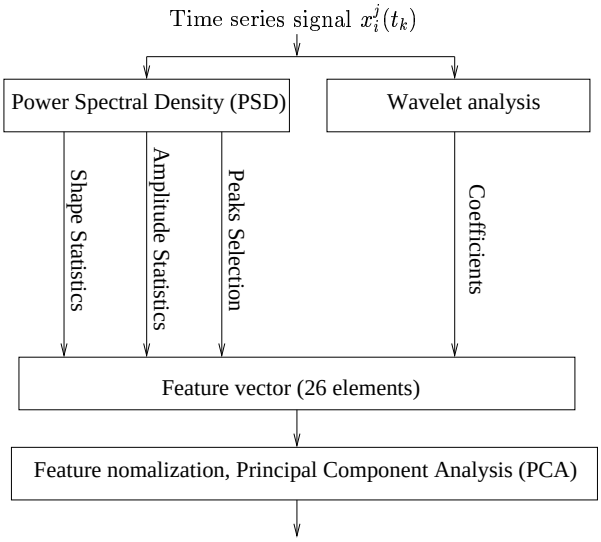


Figure 4.2: Block diagram of the feature extraction procedure.

Frequency Domain Feature Extraction

The frequency domain representation of a signal is like the trace on a spectrum analyzer, where the horizontal deflection is the frequency variable and the vertical deflection is the signal amplitude at that frequency. A time series signal $f(t)$ can be transformed into the frequency domain by using the Fourier transform, which is defined as

$$H(f) = \mathcal{F}[h(t)] = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt \tag{4.5}$$

where f is frequency and t is time. Strictly speaking, this Fourier transform representation can only be used on continuous signals, while given a series of sampled data $x(k)$, the discrete Fourier transform (DFT) is defined as

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-2\pi j \frac{nk}{N}} \tag{4.6}$$

where N is the number of samples, $x(k)$ is the time series sampled signal and $X(n)$ is the DFT representation. Since the acoustic and seismic signals emitted from ground vehicles are mainly generated by the rotation of their engines or the vibration coupled to the ground, they present unique signatures in the frequency domain for different type of targets. Therefore, the frequency domain representation is of great value in target classification using acoustic and seismic signals.

Power spectral density (PSD) is defined as the Fourier transform of the autocorrelation of the time series signal and describes the energy distribution of a signal in the frequency domain. It is defined as

$$P_{xx}(n) = \mathcal{F}[R_{xx}(k)] = \sum_{k=0}^{N-1} R_{xx}(k) e^{-j2\pi \frac{nk}{N}} \tag{4.7}$$

where $R_{xx}(k) = \frac{1}{N-k} \sum_{k=1}^{N-k} x(m)x(m+k)$ is the autocorrelation function of signal $x(m)$. Even though PSD can be calculated directly from Eq. 4.7, it is more common to use numerical methods to estimate the PSD of a signal, such as the Welch's averaged, modified periodogram method [143] for engineering use. Two 1-second samples of acoustic and seismic time series of ground vehicles and their corresponding PSD calculated using the Welch's method are shown in Fig. 2.5 and Fig. 2.11 respectively.

In the process of frequency domain feature extraction, we derive four elements of the feature vectors by calculating the higher order shape statistics and another four elements from the amplitude statistics of the PSD which provides a statistical measurement of local spectral energy content over the signal bands. Shape statistics is defined as [66]:

$$\text{Mean:} \qquad \mu_{shape} = \frac{1}{S} \sum_{k=1}^{N} kC(k)$$

$$\text{Standard deviation:} \qquad \theta_{shape} = \sqrt{\frac{1}{S} \sum_{k=1}^{N} (k - \mu_{shape})^2 C(k)}$$

$$\text{Skewness:} \qquad \gamma_{shape} = \frac{1}{S} \sum_{k=1}^{N} (\frac{k - \mu_{shape}}{\theta_{shape}})^3 C(k)$$

$$\text{Kurtosis:} \qquad \beta_{shape} = \frac{1}{S}\sum_{k=1}^{N}(\frac{k-\mu_{shape}}{\theta_{shape}})^4 C(k) - 3 \tag{4.8}$$

where $S = \sum_{k=1}^{N} C(k)$. $C(k)$ denotes the PSD magnitude for the kth frequency bin, and N denotes the number of the frequency bins. Similarly, amplitude statistics is defined as [66]:

$$\begin{aligned}
\text{Amplitude:} \qquad & \mu_{amp} = \frac{1}{N}\sum_{k=1}^{N} C(k) \\
\text{Standard deviation:} \qquad & \sigma_{amp} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(C(k)-\mu_{amp})^2} \\
\text{Skewness:} \qquad & \gamma_{amp} = \frac{1}{N}\sum_{k=1}^{N}(\frac{C(k)-\mu_{amp}}{\sigma_{amp}})^3 \\
\text{Kurtosis:} \qquad & \beta_{amp} = \frac{1}{N}\sum_{k=1}^{N}(\frac{C(k)-\mu_{amp}}{\sigma_{amp}})^4 - 3
\end{aligned} \tag{4.9}$$

The peak locations of PSD represent the dominant frequencies of the time series signal. They indicate the frequencies of the vibration of vehicles, and are suitable features for representing and classifying different targets. We choose the frequency locations of the three highest peaks and their corresponding magnitudes in PSD as six elements of the feature vector.

Time-Frequency Domain Feature Extraction

The other twelve elements of the feature vector are derived from the wavelet coefficients of the time series signal. Wavelet transform is a solid time-frequency domain signal analysis method which is designed to analyze non-stationary signals [51, 85]. The objective of wavelet transform is to represent the time series signals in terms of simple, fixed models, which are called *wavelets*. Wavelets are defined as functions that satisfy certain mathematical requirements and are used in representing data or other functions [51]. The fundamental idea behind wavelets is to analyze according to scale. Wavelets are derived from a single generating function which is called the

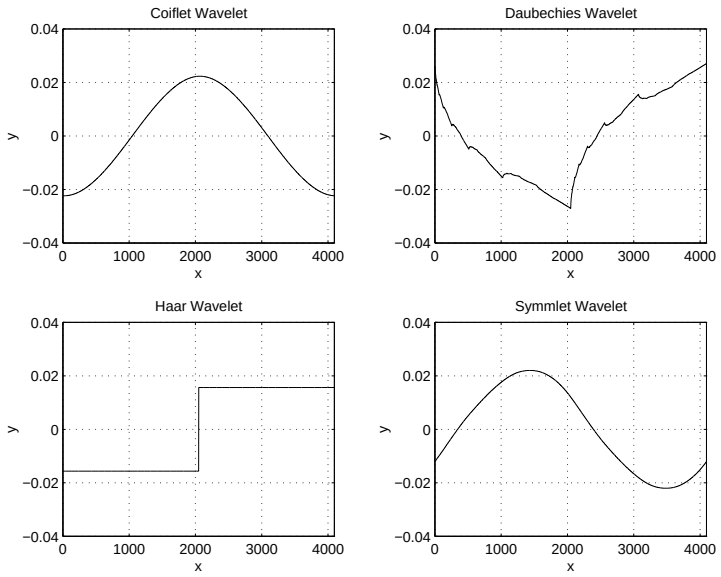


Figure 4.3: Some example mother wavelets.

mother wavelet. Generally, the mother wavelet $\psi(t)$ should satisfy the condition

$$\int \psi(t)dt = 0 \tag{4.10}$$

The translation and scaling of the mother wavelet will generate a family of real orthonormal wavelets $\psi_{a,b}(t)$ of the form [25]

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{4.11}$$

where a denotes the scaling factor and b is the translation factor. The change of a will make the wavelets cover different frequency ranges, i.e., the greater $|a|$ is, the smaller the frequency it represents [138]. The change of b corresponds to the change of the time center.

There are many different wavelet families, such as the Coiflets, the Daubechies, the Haar and the Symmlets. Some of the mother wavelets are illustrated in Fig. 4.3. Among these

wavelets, the Daubechies wavelet [36] is widely used in the engineering field since it uses a smooth finite-length kernel function to describe the details between scales of approximations which makes it suitable for solving problems in signal processing applications. The shape of the Daubechies wavelet is determined by a sequence of parameters $(c_0, c_1, \cdots, c_N)$ which satisfy the conditions

$$\sum_{n=0}^{N} c_n = \sqrt{2} \tag{4.12}$$

$$\sum_{n=0}^{N} (-1)^n n^k c_n = 0, \text{ for } k = 0, 1, \cdots, \frac{N-1}{2} \tag{4.13}$$

$$\sum_{n=0}^{N-2k} c_n c_{n+2k} = 0, \text{ for } k = 1, 2, \cdots, \frac{N-1}{2} \tag{4.14}$$

Wavelet transforms can be divided into two general categories: continuous and discrete. Continuous wavelet transform (CWT) is similar to the short time Fourier transform (STFT) in the sense that the signal is multiplied with the wavelet and the transform is computed separately for different segments of the time-domain signal. The CWT is defined as follows

$$CWT_x^{\psi}(a, b) = \frac{1}{\sqrt{|a|}} \int x(t) \psi(\frac{t-b}{a}) dt \tag{4.15}$$

where $x(t)$ denotes the time-domain signal, and $CWT_x^{\psi}(a, b)$ is the transformed signal.

Even though the CWT can be discretized and implemented using computers, this computation may take from a couple of seconds to a couple of hours depending on the signal size and the required resolution. This is because the discretized wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned [101]. This redundancy requires a significant amount of computation time and resources. Therefore, discrete wavelet transform (DWT) is defined and fast algorithms are developed in this category. In practice, the process of DWT can be implemented using a set of sampling functions which are called the *digital filter banks*. As illustrated in Fig. 4.4, the

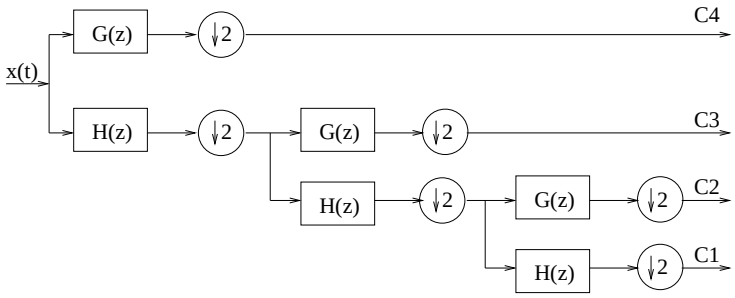


Figure 4.4: The implementation of wavelet transform.

time series signal is decomposed into wavelets using the set of digital filter banks.

We observe that the first step of wavelet transform is to input the sampled data in parallel to a low-pass filter (H) and a high-pass filter (G). The outputs of the two filters are down-sampled and kept exactly one half of the size of the input signal. After the first step, the output of the high-pass filter becomes the wavelet coefficients $C4$ at level L (in our approach, we choose $L = 3$). These coefficients represent the highest frequency wavelet level of the transform. The second step uses the output of the low-pass filter of the previous step as the input and the output of the high-pass filter of this step becomes the wavelet coefficients $C3$ at level $L - 1$. The same process continues until the level one coefficients are derived. Therefore, there are totally four series of coefficients, $C1$ to $C4$ are derived in the case $L = 3$. Figure 4.5 shows a segment of seismic signal and its wavelet coefficients. Different colors correspond to different levels of coefficients.

Given the wavelet coefficients, the average, the standard deviation and the energy of these four levels of wavelet coefficients, $C1$ to $C4$, are used as the last twelve elements in the feature vector.

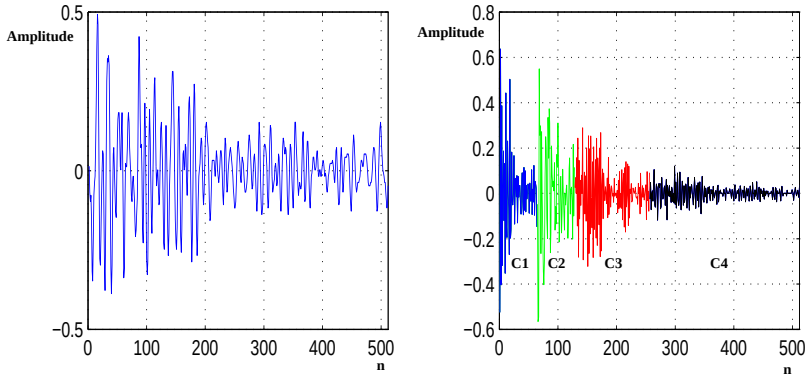


Figure 4.5: Different levels of wavelet coefficients.

Feature Normalization

Since the elements of feature vectors are derived from different domains using different methods, they are of different scale. Therefore, if the extracted feature vectors are inputed directly to the classifier, the elements with lower amplitude will be ignored. In this case, all the feature vectors need to be normalized to the same scale. The commonly used normalization method is to make the feature vectors have zero mean and unit variance, which can be expressed as [66]

$$\hat{x}(i,j) = \frac{x(i,j) - \mu_j}{\sigma_j} \tag{4.16}$$

where

$$\begin{aligned}
\hat{x}(i,j) &= \text{normalized feature element} \\
x(i,j) &= \text{original feature element} \\
\mu_j &= \frac{1}{M}\sum_{i=1}^{M} x(i,j) \\
\sigma_j &= \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(x(i,j)-\mu_j)^2} \\
i &= 1, 2, \cdots, M
\end{aligned}$$

$$
\begin{aligned}
j &= 1, 2, \cdots, N \\
M &= \text{the number of samples in the whole data set} \\
N &= \text{the number of elements in the feature vectors}
\end{aligned}
$$

Dimensionality Reduction

Since the elements of the extracted feature vectors may be correlated with each other, it is necessary to reduce the dimensionality of the feature set to a minimum but sufficient one. By doing this, it can also reduce the required amount of computation in target classification. The principal component analysis (PCA) algorithm is one of the most popularly used dimensionality reduction method in patter recognition area.

PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. Suppose the normalized feature matrix X is of dimension $M \times N$, where M is the number of observations, N is the dimension of each feature vector. The goal of PCA is to find a set of basis vectors $B = [b_1, b_2, \cdots, b_K]$ that corresponds to the maximum variance directions in the original feature space. Then the PCA transformation can be represented as

$$Y_{M \times K} = X_{M \times N} \cdot B_{N \times K} \tag{4.17}$$

where $K < N$. The basis vectors b_i should be linearly independent and orthogonal, which means

$$b_i^T b_j = \begin{cases} 1 & \text{if i = j} \\ 0 & \text{if } i \neq j \end{cases} \tag{4.18}$$

An optimal choice of the basis vectors is the eigenvectors of the feature covariance matrix. The procedure of calculating the basis vectors is as follows.

Step 1: Calculate the covariance matrix of the original feature matrix as

$$S = \frac{1}{M}(X - \mu)^T (X - \mu) \tag{4.19}$$

where μ is the mean vector of the feature set. Since the normalization process has made the feature vectors have zero-mean and unit-variance, the calculation of the covariance matrix can be simplified as

$$S = \frac{1}{M} X^T X \tag{4.20}$$

Step 2: Calculate the eigenvalues and the corresponding eigenvectors of the covariance matrix S.

Step 3: Without lose of generality, the eigenvectors are sorted in terms of their eigenvalues and those corresponding to the largest K eigenvalues are chosen to construct the transformation matrix R, i.e.,

$$R = [b_1, b_2, \cdots, b_K] \text{ of corresponding eigenvalues } \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K \tag{4.21}$$

The choice of K can be determined as

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{M} \lambda_i} \geq 1 - \eta \tag{4.22}$$

where η denotes the loss of energy.

4.2.2 Local Target Classification

After normalization and dimensionality reduction using PCA, we have extracted the most significant features that can be used as the input to the designed classifier. In our approach, the k-Nearest-Neighbor (kNN) algorithm is chosen as the classifier to categorize the targets into specific classes [43]. The basic idea of kNN is to classify a test sample $\mathbf{x}$ by assigning it to

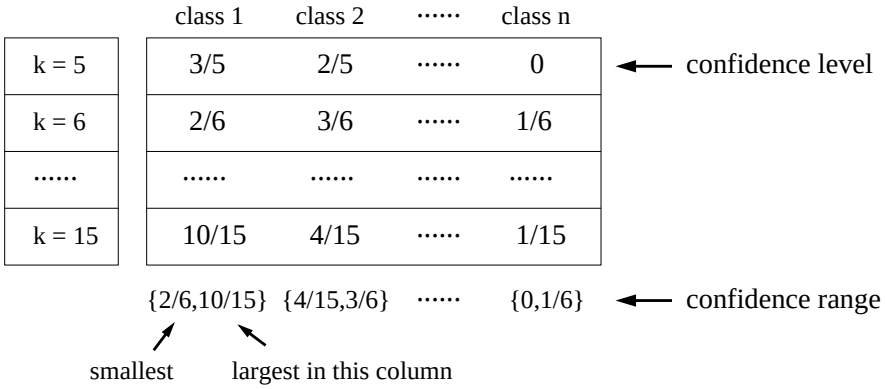


Figure 4.6: The generation of abstract intervals using modified kNN.

the class label most frequently represented among the k nearest neighbors. By using some kind of distance metric, the k nearest neighbors in the training set from the test sample $\mathbf{x}$ can be retrieved. If within this neighborhood, more samples lie in class i than any other classes, the unknown test sample is assigned as belonging to class i. The decision rule is mathematically described as the posterior probability density as:

$$p(\omega_i|\mathbf{x}) = \frac{k_i}{k} \tag{4.23}$$

In the target classification and fusion hierarchy developed in this chapter, we use a modified kNN algorithm for local signal processing to generate a classification confidence range for each class on each sensor node. Let k change in a closed-form integer range, such as $[5, 15]$. For each specific k, according to the decision rule of the kNN algorithm, classification is performed and a *confidence level* for each class is generated. Among all the confidence levels of one class for various k's, the minimal and the maximal values are chosen to form a *confidence range*, which is considered as the abstract interval for the MRI algorithm in multi-sensor fusion. The generation of the abstract interval is illustrated in Fig. 4.6.

4.3 Temporal Fusion

In the terminology of sensor network applications, the duration of time that sensors detect a changing environment is called an *event*. The signals captured at each sensing modality during the occurrence of an event might be non-stationary. In order to avoid signal processing in the context of non-stationarity which usually presents uncertainty, each event is treated as a temporal aggregation of sub-events over a short period of time. For example, in target detection applications, depending on the target's speed, each sensor node might detect a target going by as an event which lasts about 10-15 seconds. Each local target classification can then be carried out on data in a 1-second sub-event interval. In this sense, the objective of temporal fusion is to fuse all the local processing results $\{\psi_i^j(t_1), \ldots, \psi_i^j(t_k), \ldots, \psi_i^j(t_T)\}$ that correspond to the same event. Within an event, each sensor perceives the same features of its environment over time, therefore, redundant information is provided. The fusion of redundant information can reduce overall uncertainty and increase the accuracy of data interpretation.

A simple majority voting technique is adopted as the fusion function $\mathcal{F}_{temporal}$ at this level. We reason its effectiveness in fault tolerance through a general class linear combination and Lam's theorem of majority voting.

Suppose there are C possible outputs from the local processing at each sensor node. Then from all the sub-event local processing results $\{\psi_i^j(t_1), \ldots, \psi_i^j(t_T)\}$ of the same event, the occurrence of each individual possible output can be counted, which is denoted as $\{\omega_1, \ldots, \omega_C\}$. The fusion result $\overline{\psi_i^j}$ is the possible output label with the maximum occurrence:

$$\overline{\psi_i^j} = \arg\max_c \omega_c,\ c \in [1, C] \tag{4.24}$$

Majority voting is so far the simplest fusion method to implement. It does not assume prior knowledge of the behavior of each individual processing unit and does not require training on large quantities of representative results from different sources. The simple majority voting

method is a special case of a general class of linear combinations. In the context of temporal fusion, the linear combination function is denoted as

$$\overline{\psi_i^j} = \sum_{k=1}^{T} \psi_i^j(t_k) \cdot g_k \tag{4.25}$$

where $g_k = 1/T$ for simple majority voting. We consider the output of each sensor's local processing as a summation of its true discriminant function and a random noise function with zero mean, i.e., $\psi_i^j(t_k) = f(x_i^j(t_k)) + n(t_k)$, where $f(\cdot)$ denotes the true discriminant function and $n(\cdot)$ is the zero mean Gaussian noise function independent of the sensor readouts. Using simple majority voting,

$$\begin{aligned} \overline{\psi_i^j} &= \frac{1}{T}\sum_{k=1}^{T} \psi_i^j(t_k) = \frac{1}{T}\sum_{k=1}^{T}[f(x_i^j(t_k)) + n(t_k)] \\ &= \frac{1}{T}\sum_{k=1}^{T} f(x_i^j(t_k)) + E[n(t_k)] \\ &= \frac{1}{T}\sum_{k=1}^{T} f(x_i^j(t_k)) \end{aligned} \tag{4.26}$$

We observe from Eq. 4.26 that the consensus $\overline{\psi_i^j}$ only depends on the true discriminant function when the number of voters T is large enough. In this sense, the voting method actually averages out noise inference and acts as a regularizer with a smoothness assumption on the true discriminant functions.

In temporal fusion, we can also consider the local processing on each sub-event interval as an expert. Since the same local processing algorithm $\Psi(\cdot)$ is applied to all the sub-event intervals corresponding to an event, the processing results $\{\psi_i^j(t_1), \ldots, \psi_i^j(t_T)\}$ are independent and have an equal probability p of being correct. To estimate the correct rate of the consensus, we assume that each vote has only two values, correct or not, then a binomial distribution can be

used to determine the probability of the consensus being correct which is denoted as $P_C(T)$.

$$P_C(T) = \sum_{q=k}^{T} \begin{pmatrix} T \\ q \end{pmatrix} p^q (1-p)^{T-q} \tag{4.27}$$

where

$$k = \begin{cases} \frac{T}{2} + 1, & \text{if } T \text{ is even} \\ \frac{T+1}{2}, & \text{if } T \text{ is odd} \end{cases} \tag{4.28}$$

Lam proved the theorem of majority voting in [77] which gives a recursive formula describing the probability of the consensus being correct for both even and odd values of T.

Theorem 1 $P_C(2t+1) = P_C(2t) + p^{t+1}(1-p)^t \begin{pmatrix} 2t \\ t \end{pmatrix}$ *and* $P_C(2t) = P_C(2t-1) - p^t(1-p)^t \begin{pmatrix} 2t-1 \\ t \end{pmatrix}$.

From Theorem 1, several remarks on majority voting are deduced when $0 < p < 1$.

1) When the number of experts is odd ($T = 2t+1$, $T \geq 1$) and $p > 0.5$, $P_C(2t+1)$ is monotonically increasing in t and $P_C(2t+1) \to 1$ as $t \to \infty$.

2) When even numbers of experts are fused, $P_C(2t)$ is monotonically increasing if $p > t/(2t+1)$.

These conclusions coincide with the theorem of Condorcet [39] who is considered the first person to recognize the fact that the judgment of a group is superior to those of individuals provided the individuals have reasonable competence. These arguments indicate that the shorter the sub-event interval, the more votes/experts, the more probable the fused result being correct. However, shorter sub-event intervals also consume more computation power which is in conflict with the energy efficiency requirement. In Sec. 4.2, we give an application example showing how to determine this interval in a reasonable way.

4.4 Multi-Modality Fusion

Within each local sensor node, multiple sensing modalities are usually installed as each modality is only able to provide information concerning a subset of features that form a subspace in the feature space. Different modalities can compensate each other's sensing capability and provide a comprehensive view of the event. For example, the WINS (wireless integrated network sensor) nodes developed at UCLA are equipped with three channels to capture acoustic, seismic and passive infrared signals [102]. Since the signals from different channels are captured using independent devices, the signal processing results from different modalities are considered to be independent. Hence, the multi-modality fusion problem can be regarded as a classifier fusion problem and some simple classifier fusion algorithms can be implemented successfully [147]. However, due to hardware configuration and signal processing constraints, a single sensor node is usually equipped with a small number of sensing modalities (2-3 typically). In this case, the majority voting method cannot provide an unbiased estimation result. Therefore, we choose to use the Behavior-Knowledge Space (BKS) method [61, 74] as a fusion algorithm that combines the results from different sensing channels.

In the BKS method, every possible combination of the event labels from different modalities is an index to a cell in a look-up table with each entry one of the following: a *single event label* which is the one that is most often encountered among all the training samples belonging to this cell; *no label* which means there are no training samples giving the respective combination of event labels; or a set of *tied event labels* which is the case that more than one event labels has the same highest number of training samples in this cell. A 2-modality fusion example using the BKS algorithm is shown as follows.

Example: Let the number of different event labels $C = 3$, the number of modalities $M = 2$, and the number of training samples $S = 100$. A possible BKS look-up table is displayed in Table 4.1 [74].

The first column in Table 4.1 shows all the possible combinations of the 3 event labels for 2

Table 4.1: A possible BKS look-up table [74].

$\overline{\psi_i^1}, \overline{\psi_i^2}$	Occurrence of each event label	Cell label
$1, 1$	$10/3/3$	1
$1, 2$	$3/0/6$	3
$1, 3$	$5/4/5$	$1, 3$
$2, 1$	$0/0/0$	0
$2, 2$	$1/16/6$	2
$2, 3$	$4/4/4$	$1, 2, 3$
$3, 1$	$7/2/4$	1
$3, 2$	$0/2/5$	3
$3, 3$	$0/0/6$	3

sensing modalities. The numbers in the second column give the occurrence of each event label in all the training samples that correspond to a specific combination. For example, the $10/3/3$ in the first row indicates that, of all the training samples whose output from the first modality is 1 and output from the second modality is also 1, 10 samples have event label 1, 3 samples have event label 2 and 3 have event label 3. A majority voting is performed on these occurrence rates and it is decided that samples whose outputs from the two sensing modalities are $1, 1$ should have label 1. The final label is shown in the last column of Table 4.1.

The operation procedure of BKS method in multi-modality fusion is shown as Fig. 4.7, where $\overline{\psi_i^1}, \ldots, \overline{\psi_i^M}$ are the outputs of temporal fusion, also referred to as event labels, on M different channels respectively. They can be classification results (class labels), target locations (coordinates) or tracks (coordinates and directions). The BKS method is energy efficient while at the same time being fault tolerant. Although the creation of look-up table is time-consuming, for a given training set, it only needs to be done once, off-line. The multi-modality fusion process can then be carried out as a simple look-up action.

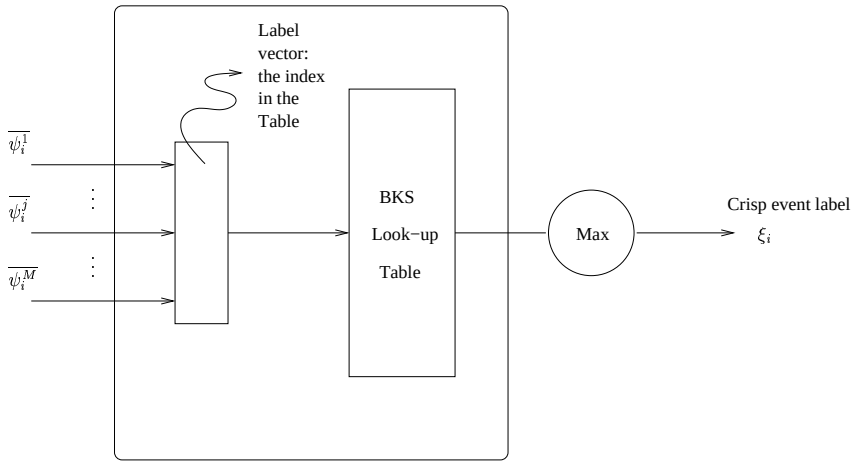


Figure 4.7: Operation of BKS method for multi-modality fusion.

4.5 Multi-Sensor Fusion Using Mobile Agent Framework

Generally speaking, a sensor network consists of a large amount of sensor nodes scattered in the field. The multiple sensor nodes can sense an event simultaneously at different geographical positions, which indicates that they can provide both redundant and complimentary information about the environment. Since the performance of an individual sensor node is very unreliable, the information derived from each sensor node needs to be fused at the highest level of the hierarchy, which is referred to as the multi-sensor fusion. In a literal sense, fusion is the process of combining data or information in such a way that the result provides more information than the sum of the individual parts [87]. Definitions of sensor data fusion employed in research literature vary in scope even though with same theme. Abidi and Gonzalez gave a comprehensive definition in [3]: "Data fusion deals with the synergistic combination of information made available by various knowledge sources such as sensors, in order to provide a better understanding of a given scene." The algorithm for multi-sensor fusion has to be robust and fault-tolerant in order to handle uncertainty and node failure. Furthermore, since the information transfer between sensor nodes consumes energy and bandwidth, the multi-sensor fusion algorithm also needs to

be energy-efficient and have the amount of data transfer minimized in the sensor network.

4.5.1 Fault Tolerance of Multi-sensor Fusion

Sensor networks are usually deployed in harsh environment to replace the role of human beings, therefore, each sensor node is subject to failure due to either lack of power, physical damage, or environmental inference. In this situation, fusion of redundant information from multiple sensor nodes is used to provide fault tolerance to sustain the sensor network functionality without interruptions of node failures. This process presents a problem to the design of sensor network since the fuser will receive multiple readings that are either partially or entirely in error. It must decide which components are faulty, as well as how to interpret at least partially contradictory readings [19].

The research of fault tolerance in multi-sensor fusion is analogous to the well-developed Byzantine generals problem (BGP). The BGP assumes a distributed decision-making process in which some participants not only make wrong decisions but maliciously attempt to force disagreement within the group [11]. As discussed in [19], if consider the BGP as a system of N independent processing elements and up to τ of them may be faulty, then certain conditions have to be satisfied in order to present correct decision through fusion: 1) The number of faulty sensors τ must be less than one third of N; and 2) τ must be less than half the connectivity of the graph. In other words, to tolerate τ faults, the system must have at least $3\tau + 1$ sensors, and every sensor must be connected directly to at least $2\tau + 1$ other sensors.

Traditionally, a sensor outputs a reading to a physical variable. This reading is prone to inaccuracies and there may be some uncertainty associated with the reading as well. In order to tolerate faulty readings using multi-sensor fusion, Marzullo proposed the concept of *abstract sensors* whose outputs are abstract intervals (bounded and connected) on the real line [88]. Following this definition, a *correct sensor* is considered as an abstract sensor whose interval estimate contains the actual value of the parameter being measured. Otherwise, it is called a

faulty sensor. Furthermore, a sensor is *tamely faulty* if it is a faulty sensor and if its output overlaps with that of a correct sensor. Using the abstract interval representation, the uncertainty of each sensor measurement can be properly modeled. If at most f faults can be tolerated among N sensors, then by taking all the intersections of $N - f$ sensor interval estimates, we are assured that the correct value of the parameter lies in one of these intersections [105].

In this chapter, we use the mobile agent framework to implement a multi-resolution integration (MRI) algorithm distributively for multi-sensor fusion.

4.5.2 Original Multi-Resolution Integration (MRI) Algorithm

The original MRI algorithm was proposed by Prasad, Iyengar and Rao in 1994 [105]. The basic idea consists of constructing a simple overlap function from the outputs of the sensors in a cluster and resolving this function at various successively finer scales of resolution to isolate the region over which the correct sensors lie [108, 109]. Each sensor in a cluster measures the same parameters. It is possible that some of them are faulty. Hence it is desirable to make use of this redundancy of the readings in the cluster to obtain a correct estimate of the parameters being observed.

We represent the multi-modality fusion result from each sensor node as the abstract interval, $\xi_n = [a_n, b_n]$ $(1 \leq n \leq N)$ where a_n and b_n are the two end points of the interval. For example, in target detection applications, a sensor can return $\xi_n = [0.6, 0.8]$ indicating that the sensor is 60% to 80% sure that there is a target in the field. The characteristic function χ_n of the nth sensor is defined as follows:

$$\chi_n(x) = \begin{cases} 1, & x \text{ is in } [a_n, b_n] \\ 0, & x \text{ is not in } [a_n, b_n] \end{cases} \tag{4.29}$$

Then the multi-sensor fusion function $\mathcal{F}_{sensor}$ can be defined as a weighted overlap function that returns the number of intervals overlapping at a certain x: $\mathcal{F}_{sensor}(x) = \sum_{n=1}^{N} \omega_n \chi_n(x)$,

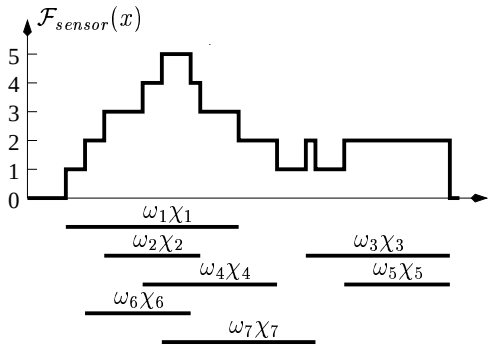


Figure 4.8: The overlap function for a set of seven sensors.

where N is the total number of sensor nodes, ω_n is the weight of the n-th sensor, and $\sum_{n=1}^{N} \omega_n = 1$. Figure 4.8 illustrates the overlap function for a set of seven sensors.

Multi-resolution analysis provides a hierarchical framework for interpreting the overlap function. Given a sequence of increasing resolutions, MRI picks the crest which is a region in the overlap function with the highest peak and the widest spread at each resolution and resolve only the crest in the next finer resolution level.

4.5.3 Distributed MRI Scheme

In a distributed sensor network, all the sensor readouts are sent to their corresponding processing elements, where the overlap function at the *finest* resolution is first generated, and the multi-resolution analysis procedure is then applied to find the crest at the *desired* resolution. With the increasing amount of sensors, the network traffic could increase dramatically.

In a mobile agent based framework, the mobile agents are dispatched from the processing center, migrating among the sensor network to collect the multi-modality fusion result from each node. Each agent always carries a *partially* integrated overlap function which is accumulated into a final version at the processing center after all the mobile agents return.

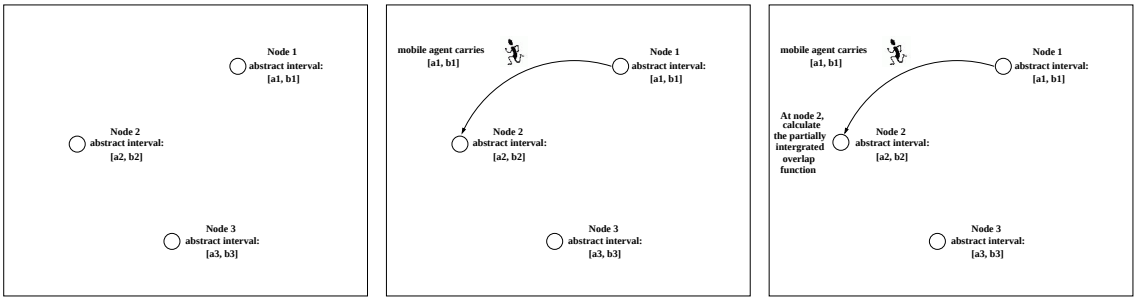
Since the carriage of partially integrated overlap function in its finest resolution counteracts

the advantages of mobile agent, the original MRI algorithm is modified so that MRI is applied *before* accumulating the overlap function. A 1-D array, serves as an appropriate data structure to represent the partially integrated overlap function carried by the agent whose size depends upon the resolution requirement. The coarser the resolution, the smaller the data buffer. The modified algorithm also provides *progressive accuracy*. When the accuracy of the fused result satisfies the requirement, the mobile agent can return to the processing center immediately without finishing the scheduled route. The mobile agent based implementation of MRI achieves the same integration result as original MRI but is more flexible, and is able to carry out the integration distributively.

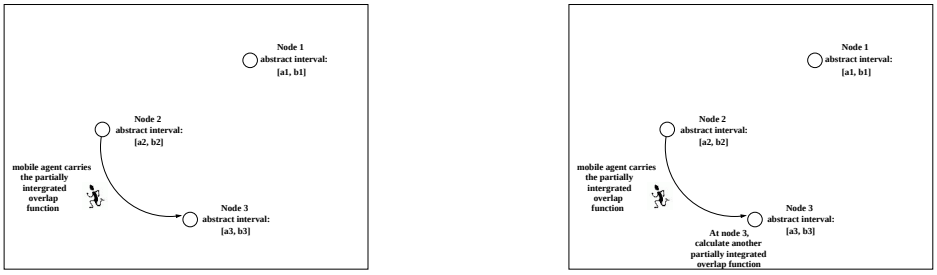
4.5.4 Mobile-agent-based Collaborative Multi-Sensor Fusion

In the developed signal and information fusion hierarchy for target classification in sensor networks, the multi-sensor fusion is at the highest level of processing. The result from each sensor node after the multi-modality fusion is a confidence range indicating the possibility that the target belongs to a specific class. Therefore, mobile agents can be employed to carry the partial integration result, migrate among the sensor nodes, and draw the final classification result.

Figure 4.9 illustrates the migration of a mobile agent in a sensor network with three sensor nodes. Each sensor node has generated an abstract interval through lower level processings of the hierarchy as shown in Fig. 4.9(a). First, the mobile agent is dispatched from node 1 and migrates to node 2 carrying $[a1, b1]$ in its buffer as in Fig. 4.9(b). When it arrives at node 2, according to the rule of MRI algorithm, a partially integrated overlap function is calculated which is shown in Fig. 4.9(c). Then the mobile agent is sent out again from node 2 to node 3 carrying the partially integrated result. When it arrives at node 3, it calculates another partially integrated result using the local abstract interval of node 3 as shown in Fig. 4.9(d) and (e). Furthermore, before sending out the mobile agent, the partially integrated overlap function can be evaluated according to a specified resolution. If it is accurate enough, the migration can be



(a) Stage 1. (b) Stage 2. (c) Stage 3.



(d) Stage 4. (e) Stage 5.

Figure 4.9: Mobile-agent-based multi-sensor fusion.

stopped immediately and the mobile agent returns to the processing center to generate the final result. Otherwise, it continues its migration in the network until the result reaches an appropriate precision or the agent finishes its pre-set itinerary.

Chapter 5

Unknown Target Recognition and Learning in Sensor Networks

In sensor network applications, a key requirement pointed out recently is to add intelligence into it so that the sensor nodes can adapt to the changing environment, handle the sensing/computation uncertainties, and accomplish complex tasks collaboratively. In the context of target recognition, this intelligence quest largely corresponds to the capability of recognizing *unknown* targets, i.e., targets without *a priori* information, and of modifying the knowledge base dynamically to incorporate the newly discovered knowledge.

Unknown target recognition remains a challenging problem in spite of the wide awareness of its importance in military surveillance, civil and environmental monitoring. Due to the fast changing property of the environment, storage space limitation, and computation burden, a single processing unit is impossible to identify all the changes and maintain a comprehensive training set to reflect these changes in the sensor network. This situation calls for each sensor node to be able to estimate the uncertainty that a given sample is generated by an unknown target and to improve the detection accuracy by combining the results from multiple sensor nodes independently derived at different physical positions.

In this chapter, a collaborative unknown target recognition algorithm is presented. It involves both local node processing and integration of local results among multiple sensors. At each local sensor node, the probability of each hypothesis that the target belongs to a specific class is estimated using a statistical rule derived from the k-Nearest-Neighbor (kNN) algorithm. The corresponding information content of different hypotheses can be represented by entropy. Intuitively, the larger the entropy, the lower the discrimination of different target classes, and the more certain that the sample is from an unknown target. An entropy-based metric is derived to measure the probability of an unknown target. Then, Dempster's rule of combination is employed to combine the estimated confidences from multiple sensor nodes since Dempster's rule has the inherent ability to incorporate decision uncertainties. If the combined confidence is above a threshold, it means there is not enough information available to decide the target type and the target is considered as unknown. Note that we assume a clustering algorithm that clusters sensor nodes based on geographical closeness has applied.

The choice of training set plays an important role in target recognition. Therefore, the training set preprocessing and its dynamic update after recognizing an unknown pattern are also discussed in this paper. To solve the skewed distribution problem induced by unknown pattern addition, a subdivision method is applied to divide the training set into subsets with the desired distribution without removing any data. This approach assures the local recognition algorithm being performed on a training set with a comparative effect of each target class.

5.1 Representing Uncertainty in Unknown Target Recognition

Unknown target recognition is a typical application example of *uncertainty reasoning*. How to represent uncertainty is application specific. Different representation schemes would affect the recognition accuracy. Halpern lists in [55] six different ways to model uncertainty, including both numeric and non-numeric. Some typical numeric representations include probability measures, Dempster-Shafer belief functions, possibility measures, and ranking functions. Some

non-numeric representations include relative likelihood on events and plausibility measures. In this paper, we employ the probability measures to represent the uncertainty of local hypotheses. In addition, we use Dempster's combination rule to resolve, to some extent, the uncertainty existed from local derivations.

5.2 Condensed Training Set Generation

In supervised classification algorithms, the selection of training set plays an important role in the performance of the resulted classifiers. A training set is composed of samples (or their corresponding feature vectors) that are representative of different target classes. Due to the imperfectness of signal acquisition and processing techniques and the complexity of real world situations, the data that construct the training set need to be refined before an effective training set can be formed.

5.2.1 Preprocessing

The main thrust of training set preprocessing is to project the training samples into another optimal basis leading to a compact and efficient target encoding in terms of recognition ability. The projection considered in this section includes two steps: principal component analysis (PCA) for dimensionality reduction and the whitening transformation.

PCA is used to eliminate the correlation between the elements of extracted feature vectors. The procedure of PCA is presented in Sec. 4.2.1. After dimensionality reduction using PCA, the lower dimensional feature set V is subjected to the whitening transformation. This procedure counteracts the fact that the Mean-Square-Error (MSE) principle underlying PCA preferentially weights low frequencies components, and generates a set of non-orthogonal bases [82]. As discussed in [37], non-orthogonal bases can lead to an over complete and robust representational space. Therefore, they can have better performance over orthogonal ones. The whitened feature

set can be calculated as

$$S = \Lambda V \tag{5.1}$$

where $\Lambda = diag\{\lambda_1^{-1/2}, \ldots, \lambda_m^{-1/2}\}$, λ_i is the ith eigenvalue of the covariance matrix of X.

5.2.2 Condensed Nearest Neighbor (CNN) Algorithm

The Condensed Nearest Neighbor (CNN) algorithm [7] is an editing procedure to eliminate the redundancy and find an optimal subset of the training set S that is small and accurate. The optimal subset Z^* is chosen out of the $2^{|S|}$ possible combinations, and the error measure is defined using regularization theory as

$$Z^* = \arg\min_Z E(Z) \tag{5.2}$$

$$E(Z) = \sum_{x \in S} L(x|Z) + \gamma|Z| \tag{5.3}$$

and

$$L(x|Z) = \begin{cases} 1, & \text{if } D(z_c, x) = \min_j D(z_j, x) \text{ and} \\ & \text{class}(x) \neq \text{class}(z_c) \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

where $z_c \in Z$ is the closest stored pattern to $x \in S$ using the distance measure $D(\cdot)$, $L(x|Z)$ is non-zero when the labels of x and z_c do not match which measures the data misfit due to error in classification, γ is the regularization parameter indicating the trade-off between the two terms in Eq. 5.3, and $\gamma|Z|$ measures the size of the subset stored and as such defines the smoothness of the class boundary. In practice, the CNN algorithm is summarized in Algorithm 2.

Algorithm 2: Condensed Nearest Neighbor algorithm.

```
Z = {};
additions = TRUE;
while additions do
    additions = FALSE;
    for all patterns in the training set do
        Randomly pick x from the training set S;
        Find z_c ∈ Z such that D(x, z_c) = min_j D(x, z_j);
        if class(x)≠class(z_c) then
            Z = Z ∪ x;
            additions = TRUE;
        end
    end
end
```

5.3 Local Unknown Target Recognition

A sensor network is usually composed of a large number of sensor nodes that integrate sensing, data processing, wireless communication, and power control components. Therefore, each sensor node has the capability to sense the environment, capture signals, and perform target recognition locally. We employ the non-parametric k-Nearest-Neighbor (kNN) algorithm [43] for target recognition at each sensor node.

5.3.1 Distance Examination

To distinguish a test sample generated by an unknown target from all the other samples that have already been stored in the training set, the simplest way is to examine the distance from this test sample to the nearest neighbor in the training set. We use this procedure as the first step in our approach for unknown target recognition. If the distance between the test sample and its nearest neighbor in the training set exceeds a threshold, i.e.,

$$D(x, z_c) > \alpha D_{\max} \tag{5.5}$$

then the sample is considered to have a probability of being unknown,

$$p(\mathcal{H}_0|x) = \begin{cases} \frac{D(x,z_c)}{D_{\max}} & \text{if } D(x,z_c) \leq D_{\max} \\ 1 & \text{otherwise} \end{cases} \tag{5.6}$$

where $\mathcal{H}_0$ denotes the hypothesis that the test sample belongs to an unknown target, $D(x, z_c)$ is the distance between the test sample and its nearest neighbor, $D_{\max}$ is the maximum distance between two samples in the training set that belong to the same class, α is a constant and we take an empirical value of 0.85.

5.3.2 Distance-based kNN Algorithm

The second step of local recognition at each sensor node is to estimate the probability of each hypothesis that the target belongs to a specific class. This task is accomplished using a statistical rule derived from the kNN algorithm. Basically, the kNN algorithm calculates the posterior probability of a test sample x belonging to a specific class based on the existing training set.

$$p(\mathcal{H}_j|x) = \frac{k_j}{k} \tag{5.7}$$

where $\mathcal{H}_j$ denotes the hypothesis that x belongs to class j, $j = 1, \cdots, C$, k_j is the number of samples in the k nearest neighbors that have a class label j.

In order to incorporate the distance measure into the classification process, we modify the classic kNN algorithm into a weighted combination procedure, i.e.,

$$p(\mathcal{H}_j|x) = \frac{\sum_{i \in k_j} \frac{1}{d_i}}{\sum_{i \in k} \frac{1}{d_i}} \tag{5.8}$$

5.3.3 Entropy-based Uncertainty Measurement

After estimating the probabilities that the test sample belongs to each class, we calculate the classification uncertainty using entropy,

$$H(uncertainty) = -\sum_{j} p(\mathcal{H}_j|x) \log p(\mathcal{H}_j|x) \tag{5.9}$$

Through the first step as shown in Eq. 5.5, we have eliminated the uncertainty due to the test sample being out of the recognition range of the given training set. Therefore, we can assume that the uncertainty $H(uncertainty)$ is generated by the vagueness of class boundaries and the introduction of unknown pattern. The larger the entropy, the more uncertain the recognition, and the more probable that the sample comes from an unknown target class. The posterior probability of hypothesis $\mathcal{H}_0$ that x is generated by an unknown target can be written as:

$$p(\mathcal{H}_0|x) = \frac{H(uncertainty)}{H_{max}} \tag{5.10}$$

where H_{max} corresponds to the entropy when the probabilities of different hypotheses $p(\mathcal{H}_j|x)$ have a uniform distribution.

5.4 Multiple Sensor Collaboration

After each sensor node performs local classification and calculates the probability that the test sample belongs to an unknown target, the decisions from different sensor nodes need to be combined at a fusion center to generate a collaborative classification result. In this step, Dempster's rule of combination is employed because it has an inherent ability to incorporate decision uncertainties from each individual source and makes it possible to assign a confidence value to a hypothesis. The algorithm of Dempster's rule of combination is depicted in Sec. 3.2.5.

Finally, a comparison between the confidence values and a predefined threshold is per-

formed. If the confidence values are above the threshold, it means there is not enough information available to determine the target type and the target is classified as unknown. The threshold is chosen empirically and in our experiment it is set to be 0.5.

5.5 Dynamic Update of the Training Set

After classification, the test samples that are classified as unknown are added to the training set. However, a major problem in this process is that the addition of unknown patterns will make the training set highly skewed, i.e., there are many more known patterns in the training set than the unknown patterns [24]. The skewness in training set distribution can make the local classifiers ineffective in giving a correct result. Therefore, the aim is to update the training set with a desired distribution (uniform distribution specifically) without removing any data.

A subdivision method is implemented to solve the problem of skewness in the training set. The training set is first divided into subsets with the desired distribution. Classification is then performed within each subset and the results from multiple classifications are combined to give the final result. For example, suppose the training set has M samples with a distribution $x : y$ ($x + y = 1$) between the number of unknown and known patterns, and $u : v$ (for example, $0.5 : 0.5$ for uniform distribution) is the desired distribution of number of samples. So the number of unknown patterns is Mx and the number of known patterns in each subset according to the desired distribution is Mxv/u. The number of subsets to be divided can be calculated as the division of the number of known patterns by the number of known patterns desired in each subset, i.e.,

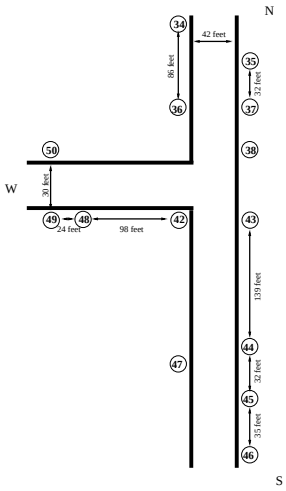
$$\frac{My}{Mxv/u} = \frac{yu}{xv} \tag{5.11}$$

Therefore, we need to divide the training set into yu/xv subsets, with each subset having Mx unknown patterns and Mxv/u known patterns. Actually, this approach is equivalent to replicating the unknown patterns across all subsets to generate the desired distribution.

Chapter 6

Experimental Results and Comparisons

According to the discussions in Chapter 3 and Chapter 4, target detection and classification tasks can be achieved distributively in sensor networks. In this chapter, experimental results of distributed detection framework and classification hierarchy are exhibited. First, three experimental scenarios are specified, which include field demos using civilian and military vehicles as targets. In Sec. 6.2, experimental results of the developed distributed source number estimation framework on two civilian vehicles are demonstrated, and the performances are compared between different approaches. Sec. 6.3 demonstrates target classification results in two field demos of military vehicles. The results from each level of processing in the proposed classification and fusion hierarchy are compared to show the effectiveness of the multi-modality, multi-sensor fusion functionality. Finally, the experimental results of the proposed unknown target recognition approach is demonstrated in Sec. 6.4.



(a) Sensor laydown.

(b) Sensoria sensor node.

Figure 6.1: The sensor laydown and the Sensoria sensor node.

6.1 Scenario Setup

6.1.1 Scenario 1 - Civilian Vehicles

The first scenario focuses on the implementation of algorithms on civilian vehicles. It is designed to capture data in a field demo that was held at BAE Systems, Austin, TX, August 2002. In this scenario, sensor nodes are deployed along the road around a T-junction, as illustrated in Fig. 6.1(a). The Sensoria WINS NG-2.0 sensor nodes are used (as shown in Fig. 6.1(b)), which consist of a dual-issue SH-4 processor running at 167MHz with 300 MIPS of processing power, RF modem for wireless communication, and up to four channels of sensing modalities, including the acoustic, seismic, and infrared. The sampling rate of both acoustic and seismic channels are 500Hz. The civilian vehicles used in this scenario include heavy diesel truck, Harley-Davidson motorcycle, pickup truck, and SUV (sport-utility vehicle), which are shown in Fig. 6.2.

(a) Diesel truck. (b) Motorcycle. (c) Pickup truck. (d) SUV.

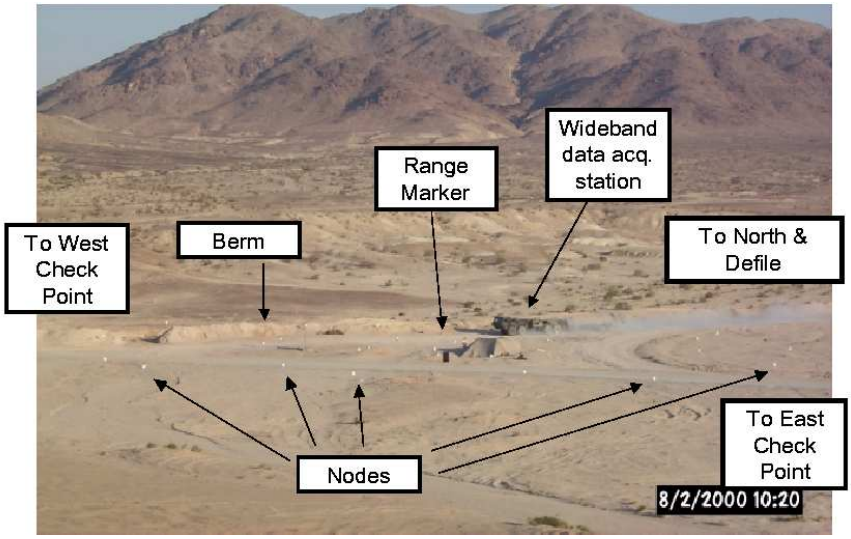
Figure 6.2: Vehicles deployed in scenario 1.

6.1.2 Scenario 2 - SITEX00 Military Ground Vehicles

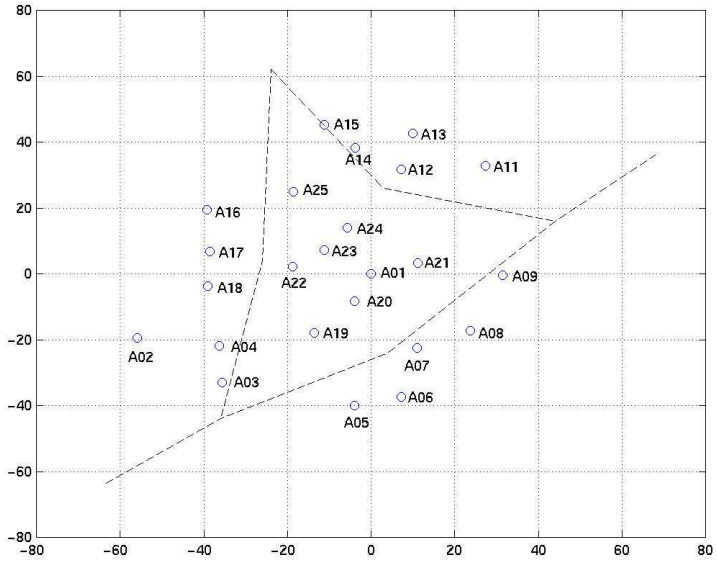
The data set in the second scenario is captured in the SITEX00 field demo that was held by DARPA SensIT (Sensor Information Technology) program at 29 Palms, CA in August, 2000. A sensor network is densely deployed at an intersection of roads, which is composed of 24 WINS NG-2.0 sensor nodes in a 7500m^2 area. The distribution of the sensor array is shown in Fig. 6.3. A typical sensor node deployment is demonstrated in Fig. 6.4. As in scenario 1, each sensor node is equipped with four sensing channels, which are acoustic, seismic, and PIRs. The sampling rate of acoustic sensing is 256Hz, while the seismic channel has a sampling rate of 128Hz. There are four military ground vehicles moving in the field, including personal owned vehicle (POV), dragon wagon (DW), light armored vehicle (LAV) and assault amphibian vehicle (AAV) as shown in Fig. 6.5.

6.1.3 Scenario 3 - SITEX02 Military Ground Vehicles

The last scenario is also designed to evaluate the performance of the proposed algorithms on military vehicles. The data set is provided by DARPA SensIT program from the SITEX02 field demo held at 29 Palms, CA in November, 2001. The nodes laydown in the field demo is shown in Fig. 6.6. Figure 6.6(a) shows the nodes laydown at the center of the intersection. Figure 6.6(b) gives the nodes laydown of the road from east to west crossing the intersection,



(a) Intersection as seen from command post (BBN Technologies).



(b) Sensor laydown.

Figure 6.3: The distribution of a sensor array at the intersection in SITEX00 field demo.

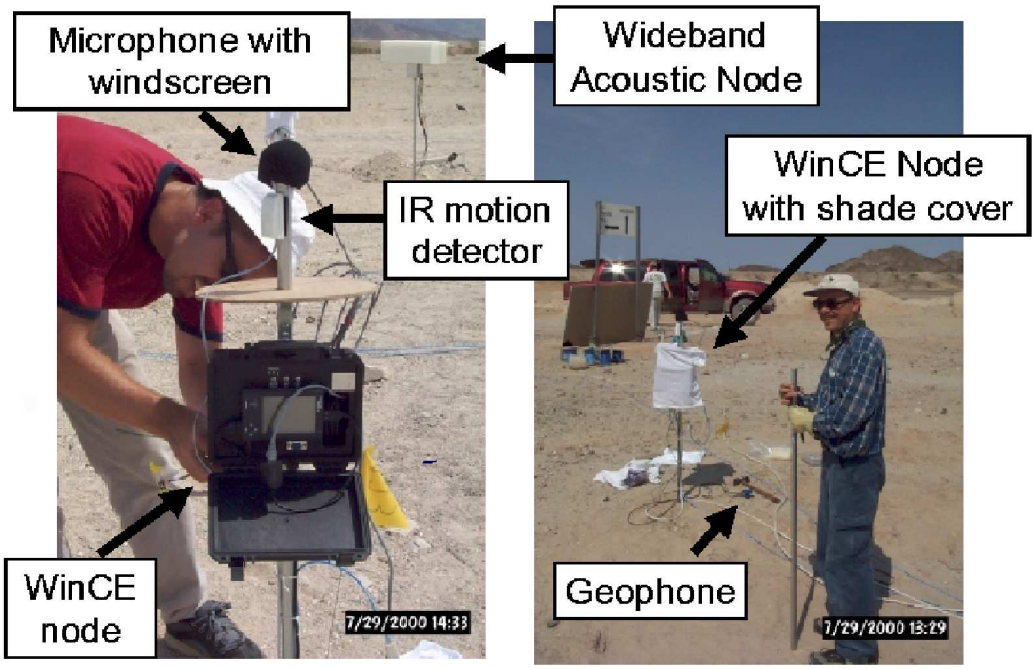
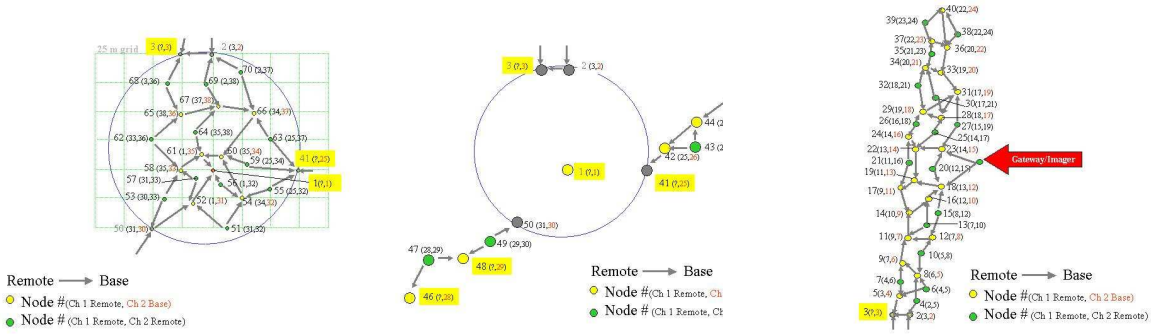


Figure 6.4: A typical sensor node deployment in SITEX00 demo.

(a) DW. (b) LAV. (c) AAV.

Figure 6.5: Vehicles deployed in scenario 2.



(a) Center (b) East-West (c) North-south

Figure 6.6: Nodes laydown of SITEX02 field demo.

with the circle indicating the center of the intersection. Figure 6.6(c) represents the nodes laydown of the road from north to south up to the intersection. A subset of sensors are chosen to form a cluster within which the signal processing algorithms are implemented. There are 3 possible target classes employed in the scenario, which are AAV, DW and HMMWV (high mobility multipurpose wheeled vehicle) as shown in Fig. 6.7. The sensor nodes are the same as employed in scenario 1 and 2, except the sampling rate of acoustic channel is increased to 1024Hz and of seismic channel to 512Hz.

6.2 Experiments on Multiple Target Detection

In order to compare the performance of different multiple target detection schemes presented in Chapter 3, we design six experiments and use data collected in scenario 1 for the evaluation. In these experiments, two civilian vehicles, a motorcycle and a diesel truck, travel along the N-S road from opposite directions and intersect at the T-junction. There are 10 nodes chosen along the road which are divided into 2 clusters of 5 sensor nodes to implement the distributed hierarchy, as illustrated in Fig. 6.8.

(a) AAV. (b) DW. (c) HMMWV.

Figure 6.7: Vehicles deployed in scenario 3.

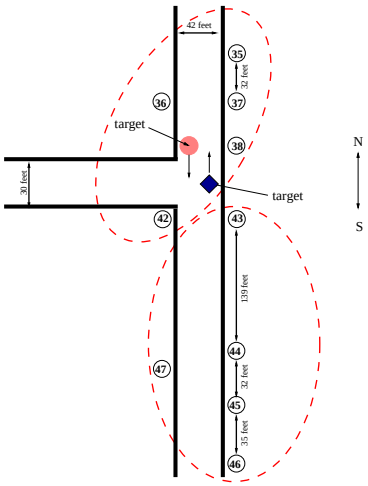


Figure 6.8: Sensor nodes clustering.

The six experiments are illustrated in Fig. 6.9 and further explained as follows,

- Experiment 1: Apply the centralized Bayesian source number estimation scheme on data collected from all the 10 sensors.
- Experiment 2: Apply the distributed estimation scheme using the centralized Bayesian source number estimation within each cluster and the Bayesian fusion method for inter-cluster posterior probability fusion.
- Experiment 3: Apply the distributed estimation scheme using the centralized Bayesian source number estimation within each cluster and the Dempster's rule of combination for inter-cluster posterior probability fusion.
- Experiment 4: Apply the progressive source number estimation scheme on all the sensors.
- Experiment 5: Apply the distributed estimation scheme using the progressive source number estimation scheme within each cluster and the Bayesian fusion method for inter-cluster posterior probability fusion.
- Experiment 6: Apply the distributed estimation scheme using the progressive source number estimation scheme within each cluster and the Dempster's rule of combination for inter-cluster posterior probability fusion.

6.2.1 Performance Metrics

We use five metrics to evaluate the performance of developed source number estimation schemes, including the average log-likelihood, the kurtosis of histogram, the detection probability, the amount of data transmission, and the energy consumption.

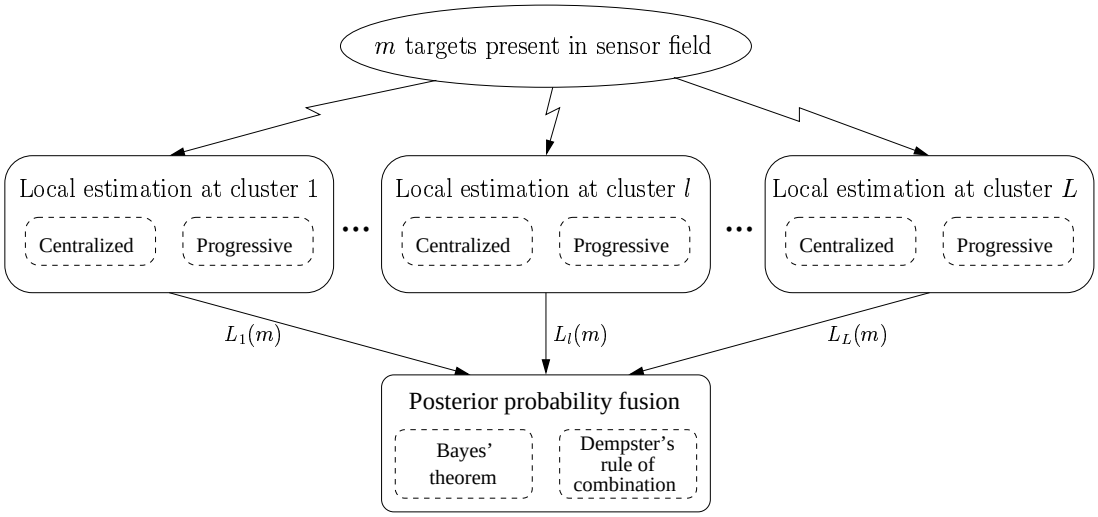


Figure 6.9: The structure of the distributed source number estimation scheme.

Average Log-likelihood

Source number estimation is basically an optimization problem in which an optimal hypothesis $\mathcal{H}_m$ is pursued that maximizes the posterior probability $P(\mathcal{H}_m|\mathbf{x}^{(t)})$ given the observation matrix. Different initialization condition might affect the optimization process at different level. In order to evaluate the performance using the log-likelihood objectively, we run the algorithm 20 times and use the *average log-likelihood* as one of the performance metrics. Figure 6.10 gives an example of the average log-likelihood calculated using the centralized scheme. The x-axis refers to different source number hypotheses $\mathcal{H}_m$, $m = 1, \cdots, 5$ and the y-axis records the corresponding average log-likelihood. From this figure, we can see that on average the correct number of targets ($m = 2$) has the largest log-likelihood compared to other source number hypotheses.

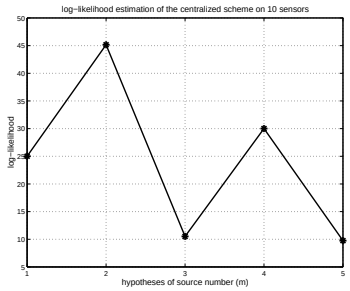


Figure 6.10: The average log-likelihood.

Kurtosis of Histogram

After executing the algorithms multiple times, a *histogram* can be generated that shows the accumulated number of occurrence that each source number hypothesis $\mathcal{H}_m$ has been chosen. In another word, the histogram shows how frequent a certain hypothesis is selected. The histogram generated using the centralized scheme is shown in Fig. 6.11 as an example. The x-axis, again, refers to different source number hypotheses and the y-axis corresponds to the number of occurrence that each hypothesis is chosen in the 20 repetitions. The histogram reveals additional information in performance analysis. Compared to Fig. 6.10, we can see that even though the hypothesis $m = 2$ has the largest average log-likelihood, it does not gain the strongest support from histogram analysis. As we can see from Fig. 6.11, the hypotheses $m = 2$ and $m = 3$ occur five times, which is even one time fewer than that of hypothesis $m = 4$. From this point of view, the histogram actually reveals the determinism and stability of a decision. We adopt *kurtosis* (β) to measure this characteristic of histogram. Kurtosis calculates the flatness of the histogram,

$$\beta = \frac{1}{C}\sum_{k=1}^{N} k(\frac{h_k - \mu}{\theta})^4 - 3 \tag{6.1}$$

where h_k denotes the value of the kth bin in the histogram, N is the total number of bins, $C = \sum_{k=1}^{N} h_k$, $\mu = \frac{1}{C}\sum_{k=1}^{N} kh_k$ is the mean, and $\theta = \sqrt{\frac{1}{C}\sum_{k=1}^{N} k(h_k - \mu)^2}$ is the variance.

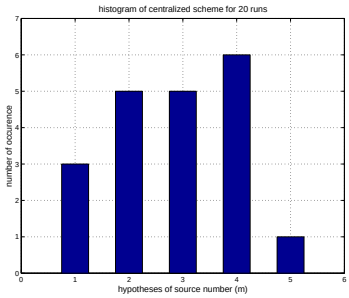


Figure 6.11: The histogram metric.

Intuitively, the larger the kurtosis, the more deterministic the approach, and the more reliable the estimate.

Detection Probability

The *Detection probability* ($P_{detection}$) is defined as the ratio between the number of correct source number estimations and the total number of estimations, i.e.,

$$P_{detection} = \frac{N_{correct}}{N_{total}} \tag{6.2}$$

where $N_{correct}$ denotes the number of correct estimations and N_{total} is the total number of estimations.

Amount of Data Transmission

In all the experiments, we assume each real number being represented as a floating point number which is 32 bits long. The *amount of data transmission* is defined as the number of bits transmitted to perform the estimation algorithm. For example, in the centralized estimation scheme, suppose there are 10 sensors deployed along the road and the algorithm is performed over 1-second segments with 500 samples each, then 144,000 bits of data need to be transmitted. While in the progressive scheme under same conditions, 37,472 bits of data (the updating

information) are transmitted.

Energy Consumption

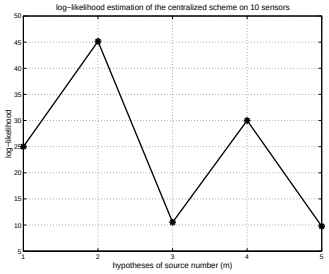
Since in a large-scale sensor network, data communication consumes most of the energy, we only consider the energy consumed on data transmission in the evaluation of *energy consumption*. According to [110], the energy consumed in data transmission can be modeled using a linear equation

$$E_{tran} = c \times size + d \tag{6.3}$$

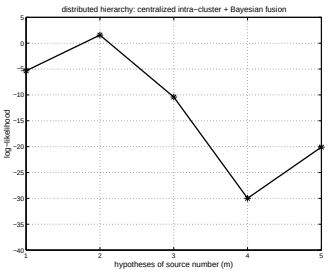
where c is a coefficient indicating the amount of energy consumed by transferring 1 byte of data, $size$ is the size of data being transferred, and d is a fixed component associated with device state changes and channel acquisition overhead. The values for c and d are different between data transmission and data receiving. Normally, we choose $c = 1.9$ and $d = 454$ for transmission while $c = 1.425$ and $d = 356$ for receiving that are measured based on a Lucent IEEE 802.11 WaveLAN PC Card using 2.4GHz direct sequence spread spectrum.

6.2.2 Experiments and Result Analysis

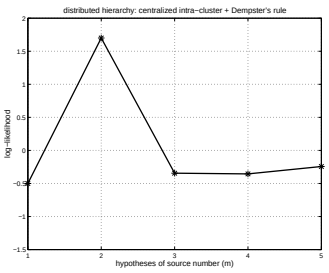
In our experiments, we test five hypotheses on the number of targets, i.e., $m = 1, 2, \cdots, 5$. All algorithms are performed on 1-second acoustic signal with a sampling rate of 500samples/second, i.e., the time instance $t = 1, 2, \cdots, 500$. The observations from sensor nodes are preprocessed component-wise to be zero-mean, unit-variance distributed. The results of the six experiments are shown in Figs. 6.12- 6.16, illustrating the average log-likelihood, the histogram, the kurtosis of histogram, the detection probability, the amount of data transmission, and the energy consumption, respectively.



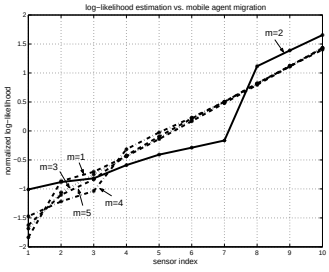
(a) Experiment 1 (centralized).



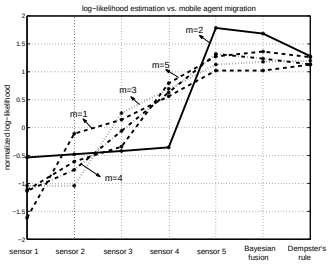
(b) Experiment 2 (centralized intra-cluster + Bayesian fusion).



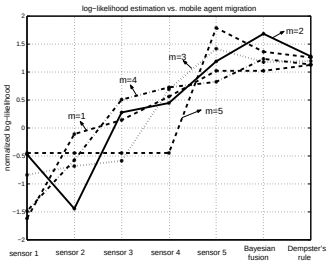
(c) Experiment 3 (centralized intra-cluster + Dempster's rule).



(d) Experiment 4 (progressive).

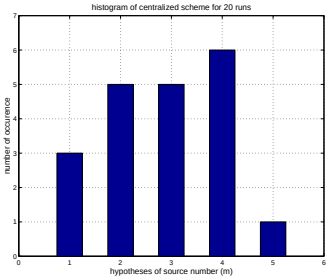


(e) Progressive estimation in cluster 1 vs. inter-cluster fusion.

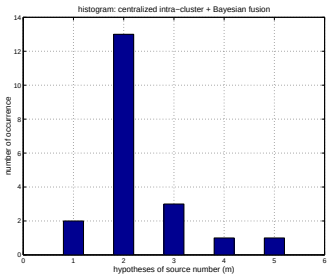


(f) Progressive intra-cluster in cluster 2 vs. inter-cluster fusion.

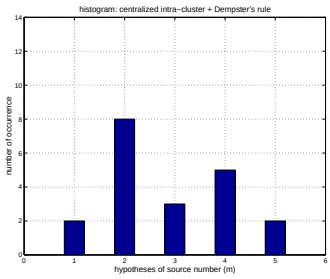
Figure 6.12: The estimated average log-likelihoods.



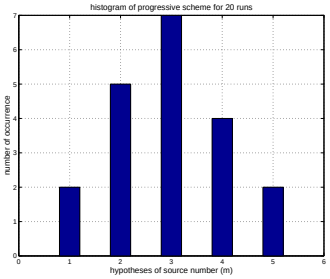
(a) Experiment 1 (centralized).



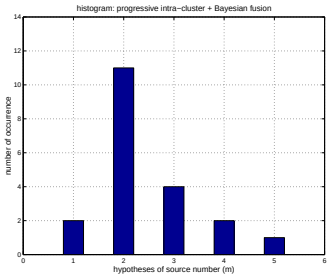
(b) Experiment 2 (centralized intra-cluster + Bayesian fusion).



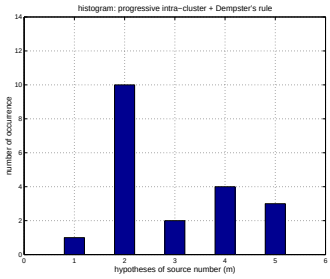
(c) Experiment 3 (centralized intra-cluster + Dempster's rule).



(d) Experiment 4 (progressive).



(e) Experiment 5 (progressive intra-cluster + Bayesian fusion).



(f) Experiment 6 (progressive intra-cluster + Dempster's rule).

Figure 6.13: The output histograms.

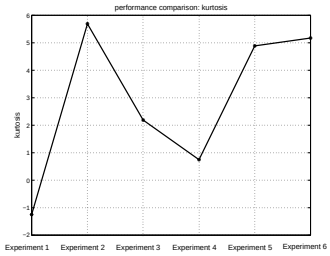


Figure 6.14: Performance comparison: kurtosis.

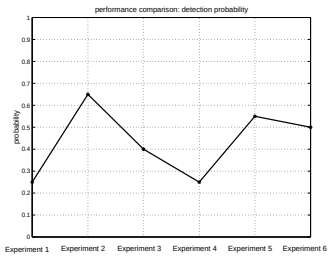
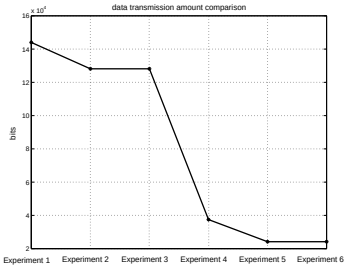
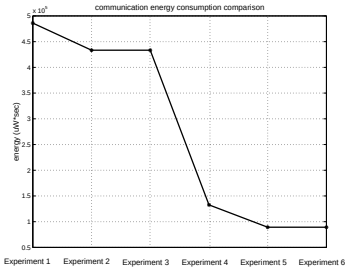


Figure 6.15: Performance comparison: detection probability.



(a) Data transmission comparison.



(b) Energy consumption comparison.

Figure 6.16: Performance comparison: amount of data transmission and energy consumption.

Comparison between Centralized and Progressive Schemes

The centralized scheme in Experiment 1 is performed at a fusion center using all the ten sensor observations. Fig. 6.12(a) shows the average log-likelihoods corresponding to different source number hypotheses and Fig. 6.13(a) displays the histogram of the occurrence of each probable number of sources when the log-likelihood function is evaluated for 20 times. From the figures we can see that even though the correct source number hypothesis ($m = 2$) has the largest average log-likelihood, this scheme does not have a stable performance in the sense that when the algorithm is performed 20 times, most of times it gives an incorrect estimate of $m = 4$.

The progressive scheme in Experiment 4 is conducted by sending a mobile agent from sensor number 38 which is at one corner of the T-junction and updating the log-likelihood estimation during its migration. Fig. 6.12(d) shows the updated average log-likelihood estimations corresponding to each stop of the mobile agent and Fig. 6.13(d) gives the histogram of source number estimation when the progressive scheme is performed for 20 times. From the log-likelihood illustration in Fig. 6.12(d), we can see that after the mobile agent migrating to the eighth sensor node, the hypothesis of true source number ($m = 2$) gains the greatest support. When the progressive scheme is performed for 20 times, it presents comparable estimation capability as the centralized approach in the sense that they give correct source number estimation the same amount of times (5 times).

One problem existed with these two approaches is that the estimation results are not consistent in multiple runs. This is partly because the algorithms are based on observations from all the sensors. If one sensor is faulty, which is very common in real sensor network applications, the estimation result will be affected.

Comparison among Different Cluster-based Distributed Estimation Schemes

There are four different ways to carry out the cluster-based distributed estimation scheme: centralized intra-cluster estimation plus Bayesian inter-cluster fusion (Experiment 2); centralized

intra-cluster estimation plus Dempster's inter-cluster fusion (Experiment 3); progressive intra-cluster estimation plus Bayesian inter-cluster fusion (Experiment 5); and progressive intra-cluster estimation plus Dempster's inter-cluster fusion (Experiment 6).

Figs. 6.12(b) and (c) illustrate the average log-likelihoods derived from Experiments 2 and 3. Figs. 6.13(b) and (c) present the corresponding histograms from Experiments 2 and 3. From the figures, we can see that in both cases, the average log-likelihood of the true number of sources $m = 2$ has the largest value. Comparing the histograms of these two distributed schemes to that of the centralized approach, the distributed approach using Bayesian inter-cluster fusion can give a correct estimation most of the times (13 out of 20), while the distributed scheme using Dempster's fusion rule can give a correct estimation 8 times and the centralized scheme 6 times only.

Fig. 6.12(e) shows the average log-likelihood estimations of each source number hypothesis corresponding to each stop of the mobile agent migration within cluster 1 (the first five points in the x-axis refers to the stops of mobile agent at the five sensor nodes within this cluster) versus the fused results using the two inter-cluster fusion methods (the last two points in the x-axis corresponds to the distributed scheme using Bayesian fusion rule and Dempster's fusion rule respectively) as designed in Experiments 5 and 6. Fig. 6.12(f) shows the average log-likelihood estimations from cluster 2 in the same manner. It can be seen that after mobile agent migrating through all the five sensor nodes, cluster 1 gives the greatest support to the true number of sources ($m = 2$) while cluster 2 gives an estimation that $m = 5$. However, after fusing the estimations from the two clusters, the Bayesian and the Dempster's fusion rule both elect the correct source number. In this way, the fusion actually increases the fidelity of the estimation. The histograms of the most probable number of sources are illustrated in Fig. 6.13(e) and (f) correspondingly. Similar to the results of Experiment 2 and 3 (centralized intra-cluster estimation plus inter-cluster fusion), the distributed scheme using progressive intra-cluster estimation and Bayesian inter-cluster fusion gives the correct source number estimation

most of times (11 out of 20), while the one with Dempster's fusion has a comparable result (10 out of 20).

Fig. 6.14 illustrates the kurtosis calculated from the six histograms shown in Fig. 6.13. It can be seen that the kurtosis of the distributed approach using centralized intra-cluster estimation and Bayesian fusion method (Experiment 2) has the largest value which is 6 times higher than the centralized scheme and 2 times higher than that using centralized intra-cluster estimation but Dempster's fusion rule. The distributed schemes using progressive intra-cluster estimation have similar kurtosis that are comparable to the largest value.

The detection probabilities are shown in Fig. 6.15. Experiment 2 again provides the highest accuracy, while the distributed schemes using progressive intra-cluster estimation have comparable performance. However, a significant advantage of the progressive source number estimation scheme is the reduction of data transmission within the network and in turn the conservation of energy.

Fig. 6.16(a) shows the amount of data transmitted through the network in all the six approaches. As we can see, in the classic centralized estimation approach (Experiment 1), 144,000 bits of data need to be transmitted, while in the progressive approach (Experiment 4), only 37,472 bits of data need to be transmitted (73.98% reduction) and in the distributed schemes using progressive intra-cluster estimation, 24,160 bits of data transmitted (83.22% reduction). Fig. 6.16(b) illustrates the corresponding energy consumption of the six approaches. We can see again that the distributed approaches using progressive intra-cluster estimation and the two inter-cluster fusion rules (Experiment 5 and 6) consume the least energy among all approaches.

6.2.3 Discussion

As demonstrated in the experiments, the cluster-based distributed approach using progressive intra-cluster estimation and Bayesian inter-cluster fusion method (Experiment 5) has the best performance in the sense that it can provide higher detection probability and more deterministic

and reliable result, while at the same time occupies less network bandwidth and consumes less energy. It seems counter-intuitive that the distributed approach performs better than the centralized approach. Here, we provide some explanations on this phenomena through experimental study as well as theoretical study.

First, in the centralized or progressive approach, the source number estimation is carried out based on observations from all the sensor nodes. If some sensors are faulty or the signals are corrupted by noise, the estimation result will be significantly affected. However, in the cluster-based distributed algorithm, the clustering approach makes sure that sensor nodes within the same cluster are close to each other. Therefore, the estimation from sensors within a cluster generally has a higher fidelity.

Second, in the derivation of the Bayesian posterior probability fusion method, the physical characteristics of sensor networks, such as the signal energy captured by each sensor node versus its geographical position, are considered, making this method more adaptive to real applications.

Third, in the multiple target detection problem, the hypotheses of different number of sources are assumed to be independent, exclusive, and form an exhaustive hypotheses space. Research has been conducted to show that in this situation, the Bayesian estimation method has a better performance. The probability of correct inference for fusion methods based on Bayesian framework and Dempster's rule of combination follows the relation as shown in Fig. 6.17 [54], which says that in a time-variant system, when the time index increases, i.e., the system goes to stable, the probability for deriving the correct fusion result favors the Bayesian method.

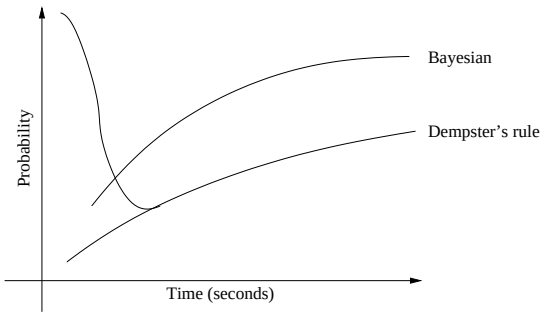


Figure 6.17: Comparison of Dempster's rule and Bayesian influence [54].

6.3 Experiments on Target Classification Hierarchy

6.3.1 SITEX00 Data Set Classification

In this experiment, we construct both the training and the testing data set from the SITEX00 database specified in Sec. 6.1.2 for multiple target classification. The distribution of sensor nodes is shown in Fig. 6.3. The classification is among four ground vehicle classes: POV, DW, LAV and AAV. We apply the classification hierarchy discussed in Chapter 4 on the data set: First, the local classification is implemented at single sensor node. The confusion matrix by using the modified kNN algorithm on sensors close to the target is shown in Table 6.1. Each row of the table indicates the numbers of actual assignments for each class. The diagonal elements are the number of correctly classified samples. We can see that the accuracy rate can be up to 92% for between-class identification.

Since in practice, we cannot expect the sensor to be always close to the target. When a sensor is further away from the target, the classification result can be severely affected. In order to improve the reliability and the accuracy of the system, we adopt multiple sensor fusion approaches. The confusion matrix by using the decentralized MRI for a sensor cluster that contains both the sensors close to the target and sensors away from the target is shown in Table 6.2.

Table 6.1: The confusion matrix of local classification.

	POV	DW	LAV	AAV	Accuracy
POV	8	1	0	3	0.67
DW	1	48	4	16	0.70
LAV	1	2	48	1	0.92
AAV	4	3	3	57	0.85

Table 6.2: The confusion matrix using the decentralized MRI on a sensor cluster.

	POV	DW	LAV	AAV	Accuracy
POV	7	0	0	1	0.88
DW	2	30	1	3	0.83
LAV	1	0	4	2	0.57
AAV	2	0	1	17	0.85

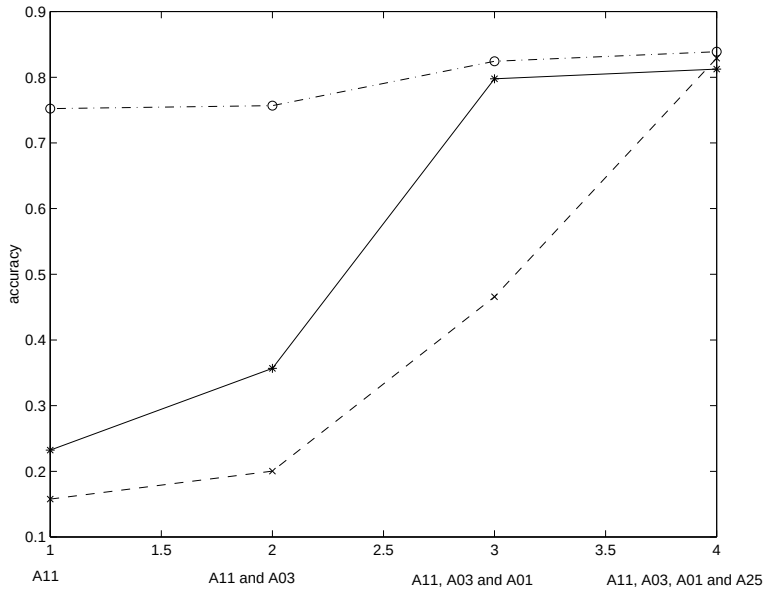


Figure 6.18: The comparison of classification accuracy for single sensor and different sensor arrays. (Solid line: target is close to A01; Dash-dot line: target is close to A11; Dashed line: target is close to A2

Figure 6.18 shows the comparison of classification accuracy by using different sensors in a sensor cluster. The target in this experiment is AAV which is a medium-size vehicle. We choose A11 as a reference sensor. We observe from the solid line that the classification accuracy by using A11 only is about 23% when the target is near A01. When we use a sensor cluster including sensors A11 and A03, the accuracy arises to about 35%. If we choose A11, A03 and A01 to construct a cluster, the accuracy is increased to about 80%. On the other hand, if the target is near A11, we can obtain an accuracy of about 75% by using A11 only, which is illustrated by the dash-dot line. The dashed line shows the situation that the target is close to sensor A25.

We can see that by using the decentralized MRI in a multiple sensor cluster, we can generally get an accuracy as high as 84% no matter where the target is in the field, whereas the result from a single sensor strongly depends on the relative position between the sensor and the target. The

MRI data fusion algorithm can also provide fault tolerance since the faults in one sensor will not affect the global result very much.

6.3.2 SITEX02 Data Set Classification Experiment

In order to assess the performance of the proposed signal and information processing hierarchy in the context of collaborative target classification, we also use the data set derived from the SITEX02 database. The nodes laydown is shown in Fig. 6.6. Three clusters of sensor nodes are chosen from the center of the intersection, the east-west road, and the north-south road with four sensor nodes each. In the field demo, there are three possible target classes, AAV, DW and HMMWV. We choose to process signals from two sensing modalities on each node, the acoustic and seismic signals, captured by microphones and geophones, respectively. A training set and a test set are generated by dividing the whole data set into three partitions, two are used as the training set and the other one is the test set.

Following the procedures of the developed signal and information processing hierarchy, firstly, the local target classification algorithm is implemented as the local processing module. Feature vectors are extracted from the observed signals over each sub-event interval and the modified kNN classifier is then performed. According to the target speed and the sensing capability of the sensors in the field demo, each event lasts about 10-15 seconds. Therefore, we choose the sub-event interval to be one second in order to satisfy the stationary requirement of most signal processing algorithms.

In the context of target classification, confusion matrix is commonly used to show the classification accuracy at each level of the signal and information processing hierarchy. In a confusion matrix, the classification results are compared with the ground truth and presented as a table, in which the horizontal labels correspond to the true labels whereas the vertical labels correspond to the classification results. Confusion matrices can also be illustrated in 3-D plots, where the x axis indicates the number of samples labeled as each target class by the classifier, and the y

axis gives the ground truth, i.e., the number of samples actually belong to each class. Hence, the diagonal columns represent the number of correctly classified samples. The flatter the non-diagonal cubes, the better the classification accuracy. Figure 6.19 - 6.21 illustrate the confusion matrices and the classification accuracy at each level of the hierarchy.

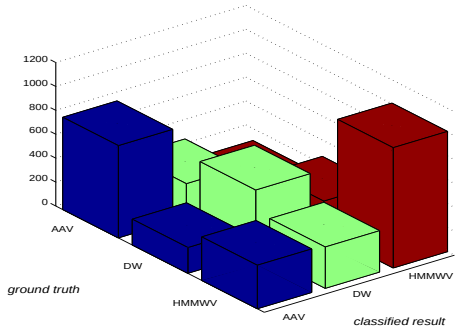
Figure 6.19(a) shows the average confusion matrix table of local target classification on 1-second acoustic sub-events and its corresponding 3-D representation. Figure 6.19(b) gives the average confusion matrix table of local classification on 1-second seismic sub-events and its 3-D plot. It is clear that in this experimental situation, the acoustic signal performs better in identifying AAV and DW, whereas the seismic signal gets a higher classification accuracy for HMMWV.

After the local target classification is performed on each sub-event interval, there are three levels of information fusion in the hierarchy. Temporal fusion is at the lowest level, which in the context of this experiment, is to fuse the classification results from all the 1-second sub-events that belong to the same event. Figure 6.19(c) shows the confusion matrix after performing the temporal fusion on all the 1-second acoustic sub-events using majority voting. Correspondingly, Fig. 6.20(a) shows the confusion matrix of temporal fusion on all the 1-second seismic sub-events.

Multi-modality fusion is then implemented to fuse the information from different sensing modalities, in this experiment, the acoustic and seismic channels. Figure 6.20(b) shows the confusion matrix after performing multi-modality fusion. It is obvious that by using the multi-modality fusion, the result can take advantage of the better performed modality and compensate the comparatively poor performed modality.

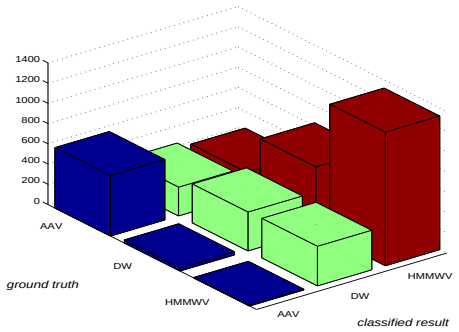
The highest level of fusion in the hierarchy is multi-sensor fusion using the mobile-agent-based MRI algorithm. After the mobile agent migrates among all the available sensors according to a predefined itinerary, an integrated result is derived using the information from each sensor node. The confusion matrix after multi-sensor fusion is shown in Fig. 6.20(c).

	AAV	DW	HMMWV	accuracy
AAV	772	283	99	0.669
DW	216	527	263	0.524
HMMWV	364	346	1006	0.586



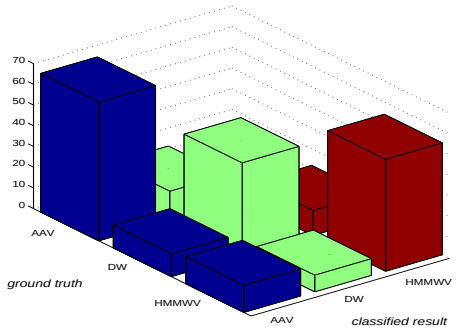
(a) Local classifi cation on 1-second acoustic sub-events

	AAV	DW	HMMWV	accuracy
AAV	597	284	233	0.536
DW	31	384	631	0.367
HMMWV	6	390	1320	0.769



(b) Local classifi cation on 1-second seismic sub-events

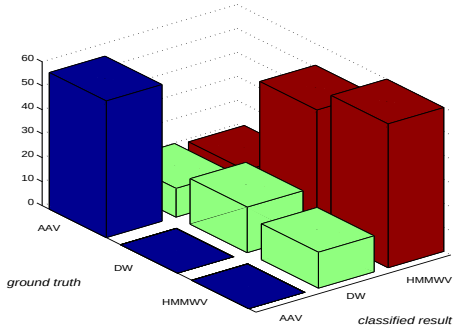
	AAV	DW	HMMWV	accuracy
AAV	67	14	1	0.8171
DW	11	45	12	0.6618
HMMWV	13	8	54	0.7200



(c) Temporal fusion on acoustic sub-events

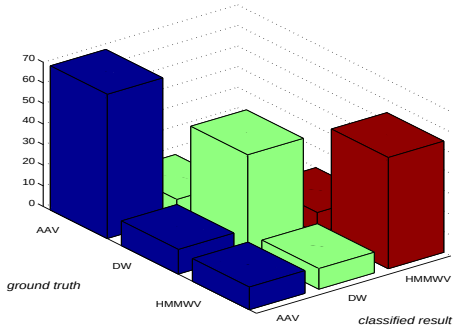
Figure 6.19: The confusion matrices at each level of the hierarchy (Part 1).

	AAV	DW	HMMWV	accuracy
AAV	57	12	11	0.7125
DW	0	19	51	0.2714
HMMWV	0	15	60	0.8000



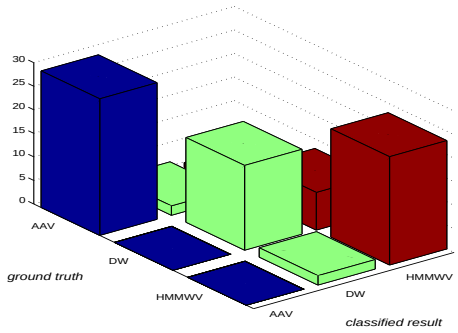
(a) Temporal fusion on seismic sub-events

	AAV	DW	HMMWV	accuracy
AAV	70	9	1	0.8750
DW	12	48	10	0.6857
HMMWV	11	10	54	0.7200



(b) Multi-modality fusion

	AAV	DW	HMMWV	accuracy
AAV	29	2	1	0.9062
DW	0	18	8	0.6923
HMMWV	0	2	23	0.9200



(c) Mobile-agent-based multi-sensor fusion

Figure 6.20: The confusion matrices at each level of the hierarchy (Part 2).

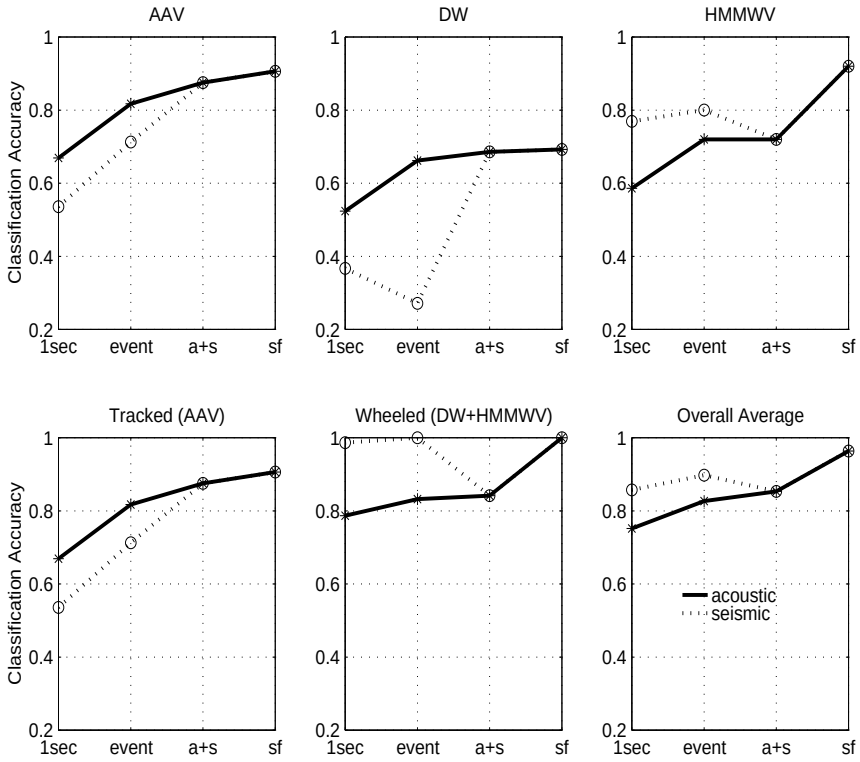


Figure 6.21: Performance evaluation of different levels in the hierarchy. Solid line: acoustic signal; Dash line: seismic signal; 1sec: averaged accuracy using 1-second sub-events; event: temporal fusion result within the same event; a+s: multi-modality fusion result; sf: multi-sensor fusion result using mobile agent.

The performance evaluation of the hierarchy for the three-class target classification example is illustrated in Fig. 6.21. In each subfigure, the solid line indicates the classification accuracy using the acoustic signal and the dash line indicates the accuracy using the seismic signal. The first data point along the x-axis is the average classification accuracy using 1-second sub-events. The second point indicates the accuracy after performing temporal fusion using majority voting. The third point shows the accuracy by performing multi-modality fusion to combine the acoustic and seismic classification results using the BKS algorithm. The last point is the accuracy using the mobile-agent-based multi-sensor fusion in the three clusters of sensor nodes.

In Fig. 6.21, the three subfigures in the first row illustrate the average classification accuracies for the three target classes, AAV, DW and HMMWV, respectively. The first two subfigures in the second row in Fig. 6.21 show the averaged accuracies for the tracked and wheeled vehicles, respectively, in which AAV is a tracked vehicle, and DW and HMMWV are wheeled vehicles. The last subfigure illustrates the overall accuracy of target classification which clearly shows improvements of the classification accuracy as the fusion hierarchy goes up.

From Fig. 6.21, we observe that, in general, the temporal fusion results are better than the local classification results on 1-second sub-events, the multi-modality fusion results are better than the temporal fusion results, and the multi-sensor fusion results are better than the multi-modality fusion results. That is, the hierarchical fusion scheme improves the classification accuracy steadily. Even though sometimes either the acoustic signal or the seismic signal can perform better than the other, the multi-modality fusion results are mostly better than using either one of them. In another word, the acoustic and seismic classification results can compensate each other through fusion. For all the three targets, the multi-sensor fusion can always achieve a better accuracy.

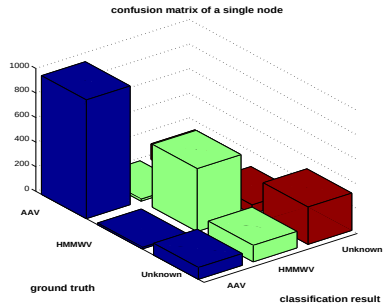
Table 6.3: Confusion matrix of a single sensor node.

	AAV	HMMWV	Unknown	accuracy
AAV	966	17	127	0.8703
HMMWV	14	512	75	0.8519
Unknown	96	134	304	0.5693

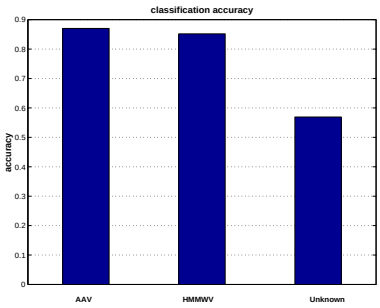
6.4 Experiments on Unknown Target Recognition

In this experiment, we use SITEX02 data set to evaluate the performance of collaborative unknown target recognition approach presented in Chapter 5. The nodes laydown in the field demo is shown in Fig. 6.6. A subset of 7 sensors are chosen to form a cluster within which the classification algorithm is implemented. In the experiment, there are three classes of military targets, AAV, DW and HMMWV, that is, $C = 3$. To perform unknown target recognition, the training set is generated using samples of two target classes (AAV and HMMWV) with 500 samples each class, while a testing set is formed including samples from all three target classes (1110, 601, and 534 samples correspondingly).

As a comparison to results of collaborative unknown target recognition, we first apply the local classification algorithm to the testing set without information of different sensor locations. In other words, we assume the testing set is generated by samples from a signal averaged sensor node. If the estimated probability that a test sample belongs to an unknown target is above the predefined threshold, the sample is assigned to be unknown. Otherwise, the class label is determined through the distance-based kNN algorithm. The average confusion matrix after local classification is shown in Table 6.3. For illustration purpose, we also show the classification accuracy in 3-D plot (Fig. 6.22(a)) and bar diagram (Fig. 6.22(b)). In the 3-D plot, a perfect target classification task should generate only three cubes along the diagonal. The higher the off-diagonal cubes, the worse the classification accuracy.



(a) Confusion matrix of a single sensor node.



(b) Classification accuracy of a single sensor node.

Figure 6.22: Classification result of a single sensor node.

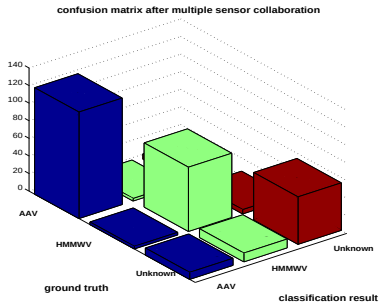
After multiple sensor collaboration, the estimated probability of hypothesis $\mathcal{H}_0$ from each sensor node are combined using Dempster's rule of combination. The combined result is compared to a threshold and the test sample is assigned to be unknown if its combined probability is below the threshold. Table 6.4 gives the classification confusion matrix after multiple sensor fusion. The same information is illustrated in Fig. 6.23.

The previous results as shown in Fig. 6.22 and Fig. 6.23 are computed according to the classification of the three target classes. However, in order to evaluate the proposed unknown target recognition approach, we examine the confusion matrix generated by only considering two categories: known and unknown targets. Figure 6.24 illustrates the confusion matrix and the classification accuracy of discriminating known and unknown patterns. Its corresponding ROC curve is shown in Fig. 6.25.

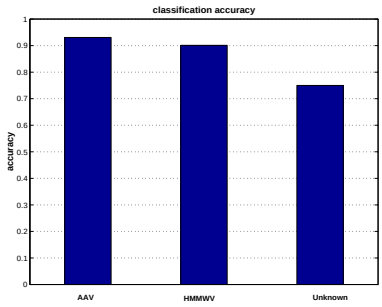
ROC (Receiver Operating Characteristic) curve is a useful tool in representing the performance of a particular classifier as some parameter is varied over its range. The four axes in Fig. 6.25 correspond to four measurements: true positive (the object is unknown and the classifier says it is unknown), true negative (the object is known and the classifier says it is known),

Table 6.4: Confusion matrix after multiple sensor collaboration.

	AAV	HMMWV	Unknown	accuracy
AAV	121	3	6	0.9308
HMMWV	3	73	5	0.9012
Unknown	8	10	54	0.75

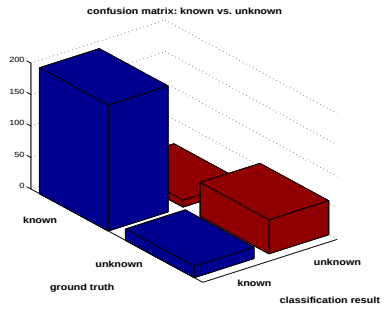


(a) Confusion matrix after multiple sensor collaboration.

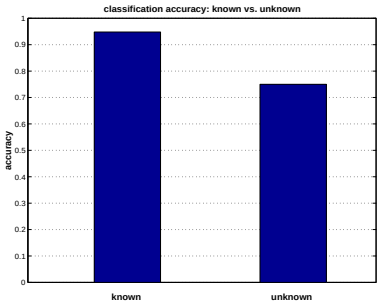


(b) Classifi cation accuracy after multiple sensor collaboration.

Figure 6.23: Classification result after multiple sensor collaboration.



(a) Confusion matrix: known vs. unknown patterns.



(b) Classification accuracy: known vs. unknown patterns.

Figure 6.24: Classification result: known vs. unknown patterns.

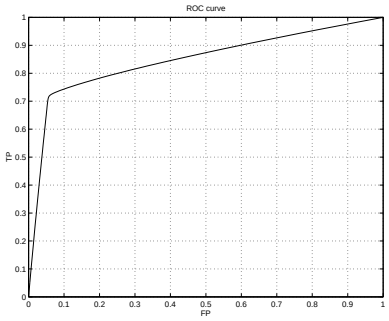


Figure 6.25: ROC curve: known vs. unknown patterns.

false negative (the object is unknown and the classifier says it is known), and false positive (the object is known and the classifier says it is unknown). Generally, the ROC curve with the sharpest bend, i.e., which passes the closest to the upper left corner, and the one with the smallest area above it has the best performance.

Chapter 7

Conclusions and Future Work

In this dissertation, we have described approaches on collaborative multiple target detection and classification in sensor networks.

7.1 Summary of Contributions

The problem of target detection in sensor networks can be discussed from two avenues: single target detection and multiple target detection. In general, single target detection can be achieved using a simple energy detector. For multiple target detection, especially when the targets are close to each other, the observations from the sensor nodes will be linear/nonlinear combinations of the source signals. Under the assumption that the number of sensors is greater than the number of sources, an ICA-based source number estimation approach can be used to dynamically estimate the number of targets in a specific sensing field. However, due to the sheer amount of sensors deployed, the limited wireless communication bandwidth, and the battery-powered fact of each sensor node, classic centralized approach would not provide satisfactory solution. Therefore, a progressive source number estimation algorithm is presented utilizing the iterative relationship between sensors and implemented using a mobile agent framework. Since each sensor node has a limited capability in sensing and communicating only within a certain

range, a cluster-based distributed estimation scheme is further developed where a local source number estimation is performed within each cluster using either the centralized or progressive approach, and a probability fusion method is then applied to combine the local estimations to generate a global decision. A posterior probability fusion method based on Bayes' theorem is derived, and for comparison, the Dempster's rule of combination is implemented.

Sensor networks usually have thousands of sensor nodes deployed densely in the field, where each sensor node can be equipped with multiple sensing modalities. When an event is detected, the multiple sensor nodes can collaborate with each other to achieve better performance and higher reliability. In this dissertation, a multi-modality, multi-sensor fusion hierarchy is proposed for collaborative target classification in sensor networks. The hierarchy is composed of four levels of enabling algorithms: local signal processing, temporal fusion over each sensing modality, multi-modality fusion at each sensor node, and multi-sensor fusion across a cluster of sensor nodes using a mobile agent framework. When dispatched, the mobile agents migrate among the sensor network and integrate the local results. Therefore, each mobile agent always carries a partially integrated result which is accumulated into a final version at the processing element after all the mobile agents return. In this way, the network bandwidth requirement is reduced and the network scalability and stability can be improved.

In order to handle the uncertainty in target recognition and make the information processing adaptive to changing environment, a collaborative approach is presented for unknown target recognition in sensor networks. This approach involves both local node processing and integration of local results among multiple sensors. At each local sensor node, after training set preprocessing, target classification is performed using the distance-based kNN algorithm and the probability of each hypothesis that the test sample belongs to a specific target class is estimated. In order to calculate the probability of unknown, an entropy-based metric is introduced according to the fact that the larger the entropy, the less discriminative the classification, and the larger the probability of unknown. Then Dempster's rule of combination is applied to combine

the estimated probabilities from multiple sensor nodes. If the fused confidence is beyond a predefined threshold, we assign the corresponding sample as unknown. Finally, the training set is updated by adding in the unknown target samples. The updated training set is then subdivided into subsets with desired distribution.

Experiments are carried out on three data sets, including real data collected from field demos with both civilian and military vehicles as targets. The experimental results show that the collaborative target detection algorithm and the classification hierarchy we developed exhibit good performance and high reliability.

7.2 Directions for Future Research

The ideas and concepts in this dissertation offer promising solutions to the problem of distributed information processing in sensor networks, they also suggest several interesting avenues for future research.

- **Intelligence incorporation.** In this dissertation, algorithms for collaborative multiple target detection and recognition in sensor networks have been developed. It is attractive to incorporate higher level intelligence into sensor network applications to provide the global assessment in the field. In order to obtain a high-level understanding (awareness) of the situation, data need to be disseminated with respect to the environment, the hidden relationship among entities be evaluated, and the patterns in time and space be extracted.

- **Performance optimization.** With the fast approaching of digital electronics technology and wireless communications, more powerful sensor nodes ("smart sensors") have been presented that integrate sensing, communication, and processing capabilities together. In response to the hardware improvement, the application-specific algorithms need to evolve to deal with dynamically changing environment and to comply with the hardware performance. Innovative computing paradigms are also needed to further improve the perfor-

mance as a whole. In this dissertation, agent-based distributed information processing has been used to handle uncertainty and node failure, to reduce the amount of communication and energy consumption. Different applications have shown that agent-based distributed processing is a promising approach in sensor networks.

- **Software-hardware integrated system development in sensor networks.** Since the ultimate goal of a sensor network is to solve a specific problem in real world, hardware platform design is the foundation while software development and implementation is the essence. The development of an integrated system with both hardware and software components is desired to accomplish complex tasks efficiently. The processing within a sensor network should be transparent to human operators, which means given a query, the integrated sensor system should be able to operate autonomously and return the result.

Bibliography

Bibliography

[1] *International Workshop on Mobile Agent Development and Applications*, Orlando, FL, July 2002.

[2] K. Aberer and M. Hauswirth. An overview on peer-to-peer information systems. In *Proceedings of Workshop on Distributed Data and Structures*, Paris, France, 2002.

[3] M. A. Abidi, Rafael C. Gonzalez, and Ralph C. Gonzalez, editors. *Data fusion in robotics and machine intelligence*. Academic Press, 1992.

[4] J. Agre and L. Clare. An integrated architecture for cooperative sensing networks. *Computer*, 33(5):106–108, May 2000.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[7] E. Alpaydin. Voting over multiple condensed nearest neighbors. *Artificial Intelligence Review*, 11:115–132, 1997.

[8] H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 21–30, 1999.

[9] A. Averbuch, E. Hulata, V. Zheludev, and I. Kozlov. A wavelet packet algorithm for classification and detection of moving vehicles. *Multidimensional Systems and Signal Processing*, 12(1):9–31, 2001.

[10] S. Baker. CORBA implementation issues. *IEE Colloquium (Digest)*, (007):5/1–5/3, January 1994.

[11] M. Barborak, M. Malek, and A. Dahbura. The consensus problem in fault tolerant computing. *ACM Computing Surveys*, pages 171–220, June 1993.

[12] BBNT. SensIT august '00 experiment plan. Technical report, DARPA SensIT Program, 2000.

[13] A. J. Bell and T. J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[14] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: embedded mobile networking. *IEEE Personal Communications*, 4(5):8–15, October 1997.

[15] A. D. Birrell and B. J. Nelson. Implementing remote procedure calls. *ACM Transactions on Computer Systems*, 2(1):39–59, February 1984.

[16] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[17] R. Braunling, R. M. Jensen, and M. A. Gallo. Acoustic target detection, tracking, classification, and location in a multiple target environment. In *Proceedings of SPIE: Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors*, volume 3081, pages 57–66, April 1997.

[18] R. R. Brooks. Reactive sensor networks. `http://strange.arl.psu.edu/RSN/`.

[19] R. R. Brooks and S. S. Iyengar. Robust distributed computing and sensing algorithm. *Computer*, 29(6):53–60, June 1996.

[20] R. R. Brooks and S. S. Iyengar. *Multi-sensor fusion: fundamentals and applications with software*. Prentice Hall, Inc, New Jersey, 1997.

[21] J. M. Caicedo, E. Clayton, S. J. Dyke, and M. Abe. Structural health monitoring for large structures using ambient vibrations. In *Proceedings of the ICANCEER Conference*, Hong Kong, August 2002.

[22] M. J. Caruso and L. S. Withanawasam. Vehicle detection and compass applications using AMR magnetic sensors. Technical report, Honeywell, SSEC, May 1999.

[23] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. In *2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, April 2001.

[24] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, pages 67–74, November/December 1999.

[25] B. H. Chen, X. Z. Wang, S. H. Yang, and C. McGreavy. Application of wavelets and neural networks to diagnostic system development. *Computers and Chemical Engineering*, 23:899–906, 1999.

[26] J. C. Chen, K. Yao, and R. E. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, 19(2):30–39, March 2002.

[27] H. C. Choe, R. E. Karlsen, G. R. Gerhert, and T. Meitzler. Wavelet-based ground vehicle recognition using acoustic signal. In *Proceedings of SPIE: Wavelet Applications III*, volume 2762, pages 434–445, April 1996.

[28] C. Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.

[29] R. Choudrey, W. D. Penny, and S. J. Roberts. An ensemble learning approach to Independent Component Analysis. In *Proceedings of Neural Networks for Signal Processing*, Sydney, December 2000.

[30] J. Claassen, M. Ladd, and G. Elbring. The feasibility of monitoring continuous wave sources with seismic arrays. In *Proceedings of SPIE: Unattended Ground Sensor Technologies and Applications*, volume 3713, pages 22–32, April 1999.

[31] GotDotNet Community. About mobile code. `http://www.gotdotnet.com/team/clr/about\_mobilecode.aspx`.

[32] P. Comon. Independent component analysis, a new concept. *Signal Processing*, 36(3):287–314, April 1994.

[33] Alliance Consulting. What is peer-to-peer? `http://www.peertohere.com`.

[34] Sensoria Corporation. sGate developer's platform. `http://www.sensoria.com/sgate.html`.

[35] Crossbow: smarter sensors in silicon. `http://www.xbow.com/Products/Wireless\_Sensor\_Networks.htm`.

[36] I. Daubechies. *Ten lectures on wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1992.

[37] J. G. Daugman. *Computational Neuroscience*, chapter An information-theoretic view of analog representation in striate cortex, pages 403–424. MIT Press, 1990.

[38] A. D'Costa and A. M. Sayeed. Collaborative signal processing for distributed classification in sensor networks. In *The 2nd International Workshop on Information Processing in Sensor Networks*, Palo Alto, CA, April 2003.

[39] N. C. de Condorcet. *Essai sur l'Application de l'Analyze à la Probabilité des Décisions Rendues à la Pluralité des Voix*. Imprimérie Royale, Paris, France, 1785.

[40] K. A. Delin and S. P. Jackson. Sensor web for in situ exploration of gaseous biosignatures. In *Proceedings of 2000 IEEE Aerospace Conference*, Big Sky, MT, March 2000.

[41] `http://hyperphysics.phy-astr.gsu.edu/hbase/sound/dopp.html`.

[42] H. Doyle. *Seismology*. Wiley, 1995.

[43] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2nd edition, 2001.

[44] I. A. Erteza, G. J. Elbring, T. S. McDonald, and J. P. Claassen. Case study: unattended ground sensor phenomenology and signal processing. In *Proceedings of SPIE: Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors*, volume 3081, pages 88–98, July 1997.

[45] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[46] D. Estrin, M. Srivastava, and A. Sayeed. Wireless sensor networks. In *Mobicom 2002 Tutorial T5*, 2002.

[47] National Science Foundation. Sensors and sensor networks. `http://www.nsf.gov/pubs/2003/nsf03512/nsf03512.htm`, 2003.

[48] Introduction to geophysical exploration. `http://www.mines.edu/fs\_home/tboyd/GP311`.

[49] C. Ghezzi and G. Vigna. Mobile code paradigms and technologies: a case study. In *Proceedings of the First International Workshop on Mobile Agents*, pages 39–49, Germany, April 1997.

[50] Glolab. How infrared motion detector components work. `http://www.glolab.com/pirparts/infrared.html`.

[51] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995.

[52] ARC Group. Wireless ITS - intelligent transportation systems signaling, traffic control and information systems, 2001.

[53] L. J. Guibas. Sensing, tracking, and reasoning with relations. *IEEE Signal Processing Magazine*, 19(2):73–85, March 2002.

[54] D. L. Hall. *Mathematical techniques in multisensor data fusion*. Artech House, Inc., 1992.

[55] J. Y. Halpern. *Reasoning about uncertainty*. The MIT Press, 2003.

[56] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless micro sensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 3005–3014, 2000.

[57] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Mobile Computing and Networking*, pages 174–185, 1999.

[58] J. Herault and J. Jutten. Space or time adaptive signal processing by neural network models. In J. S. Denker, editor, *Neural Networks for Computing: AIP Conference Proceedings 151*, New York, 1986. American Institute for Physics.

[59] How do microphones work, and why are there so many different types? http://electronics.howstuffworks.com/question309.htm.

[60] T. T. Hsieh. Using sensor networks for highway and traffic applications. *IEEE Potentials*, pages 13–16, April/May 2004.

[61] Y. S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–93, January 1995.

[62] A. Hyvarinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[63] A. Hyvarinen and E. Oja. Independent component analysis: a tutorial. http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/, April 1999.

[64] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[65] D. Keck. The Doppler effect. http://www.eritreaplanet.com/data/sl1/0008.htm.

[66] D. H. Kil and F. B. Shin. *Pattern recognition and prediction with application to signal characterization*. American Institute of Physics, 1996.

[67] L. E. Kinsler, A. B. Coppens, A. R. Frey, and Sanders J. V. *Fundamentals of acoustics*. J. Wiley and Sons, 4th edition, December 1999.

[68] A. N. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. Master's thesis, Massachusetts Institute of Technology, June 2000.

[69] K. H. Knuth. A Bayesian approach to source separation. In *Proceedings of First International Conference on Independent Component Analysis and Blind Source Separation: ICA'99*, pages 283–288, 1999.

[70] T. Kohonen. The self-organizing map. *Neurocomputing*, 21:1–6, November 1998.

[71] J. Kumagai. The secret life of birds. *IEEE Spectrum*, 41(4):42–49, April 2004.

[72] S. Kumar and D. Shepherd. Sensit: sensor information technology for the warfighter. In *Proceedings of 4th International Conference on Information Fusion*, pages TuC1–3–TuC1–9, 2001.

[73] S. Kumar, D. Shepherd, and F. Zhao. Collaborative signal and information processing in micro-sensor networks. *IEEE Signal Processing Magazine*, 19(2):13–14, March 2002.

[74] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34:299–314, 2001.

[75] The Edgerton Center Corridor Lab. `http://web.mit.edu/Edgerton/www/Optical`.

[76] D. E. Lake. Harmonic phase coupling for battlefield acoustic target identification. In *Proceedings of 1998 International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2049–2052, Piscataway, NJ, 1998.

[77] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(5):553–568, September 1997.

[78] D. B. Lange and M. Oshima. Seven good reasons for mobile agents. *Communications of the ACM*, 42(3):88–89, March 1999.

[79] T. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowski. A unifying information-theoretic framework for independent component analysis. *International Journal on Mathematical and Computer Modeling*, 1999.

[80] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed. Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine*, 19(2):17–29, March 2002.

[81] R. Linsker. Local synaptic learining rules suffice to maximize mutual information in a linear network. *Neural Computation*, 4:691–702, 1992.

[82] C. Liu and H. Wechsler. Face recognition using evolutionary pursuit. In *Proceedings of the Fifth European Conference on Computer Vision*, pages 596–612, June 1998.

[83] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):901–931, September/October 1989.

[84] D. J. C. MacKay. Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 175–204. Kluwer, 1999.

[85] S. Mallat. *A wavelet tour of signal processing*. New York: Academic, 2nd edition, 1999.

[86] Y. Mallet, D. Coomans, J. Kautsky, and O. De Vel. Classification using adaptive wavelets for feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1058–1066, October 1997.

[87] J. Manyika and H. Durrant-Whyte. *Data fusion and sensor management: a decentralized information-theoretic approach*. Ellis Horwood, 1994.

[88] K. Marzullo. Tolerating failures of continuous-valued sensors. *ACM Transactions on Computer Systems*, 8(4):284–304, 1990.

[89] R. McQuillin, M. Bacon, and W. Barclay. *An introduction to seismic interpretation*. Gulf Publishing Company, 1984.

[90] `http://hyperphysics.phy-astr.gsu.edu/hbase/audio/`.

[91] B. Nickerson and R. Lally. Development of a smart wireless networkable sensor for aircraft engine health management. In *Proceedings of 2001 Aerospace Conference*, volume 7, pages 3255–3262, 2001.

[92] C. E. Nishimura and D. M. Conlon. Iuss dual use: monitoring whales and earthquakes using sosus. *Marine Technology Society Journal*, 27(4):13–21, 1994.

[93] Optical radiation detectors. `http://elchem.kaist.ac.kr/vt/chem-ed/optics/detector/detector.htm`.

[94] J. M. Ortolf. Research support the unattended ground sensor mission. In *Proceedings of SPIE: Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors*, volume 3081, pages 42–49, July 1997.

[95] B. J. Overeinder, E. Posthumus, and F. M. T. Brazier. Integrating peer-to-peer networking and computing in the agent scape framework. In *Proceedings of the Second International Conference on Peer-to-Peer Computing (P2P'02)*, pages 96–103, Sweden, September 2002.

[96] T. A. Pering, T. D. Burd, and R. W. Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proceedings on ISLPED*, pages 76–81, 1998.

[97] S. Phoha and D. S. Friedlander. Semantic information fusion. DARPA SensIT PI Meeting, October 2000.

[98] `http://www.o-eland.com/active/source/photodiode_101.htm`.

[99] K. Pister. My view of sensor networks in 2010. http://www.eecs.berkeley.edu/~pister/SmartDust/in2010.

[100] K. Pister. Smart Dust: autonomous sensing and communication in a cubic millimeter. http://robotics.eecs.berkeley.edu/~pister/SmartDust/, 2001.

[101] R. Polikar. The wavelet tutorial. http://engineering.rowan.edu/polikar/WAVELETS/WTtutorial.html, 1999.

[102] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[103] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, 19(2):51–60, March 2002.

[104] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): design and construction. In *Proceedings on IEEE Data Compression Conference*, pages 158–167, Snowbird, UT, March 1999.

[105] L. Prasad, S. S. Iyengar, and R. L. Rao. Fault-tolerant sensor integration using multiresolution decomposition. *Physical Review E*, 49(4):3452–3461, April 1994.

[106] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[107] H. Qi, S. S. Iyengar, and K. Chakrabarty. Distributed sensor networks - a review of recent research. *Journal of the Franklin Institute*, 338:655–668, 2001.

[108] H. Qi, S. S. Iyengar, and K. Chakrabarty. Multi-resolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, 31(3):383–391, August 2001.

[109] H. Qi, X. Wang, S. S. Iyengar, and K. Chakrabarty. High performance sensor integration in distributed sensor networks using mobile agents. *International Journal of High Performance Computing Applications*, 16(3):325–335, 2002.

[110] H. Qi, Y. Xu, and X. Wang. Mobile-agent-based collaborative signal and information processing in sensor networks. *Proceedings of IEEE, Special Issue on Sensor Networks and Applications*, 91(8):1172–1183, August 2003.

[111] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[112] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey on sensor networks. University of Texas at Dallas, Course report for Mobile Computing (CS 6392).

[113] S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, series B*, 59(4):731–758, 1997.

[114] S. Roberts and R. Everson, editors. *Independent Component Analysis: Principles and Practice*. Cambridge University Press, 2001.

[115] S. J. Roberts. Independent component analysis: source assessment & separation, a bayesian approach. *IEE Proceedings on Vision, Image, and Signal Processing*, 145(3):149–154, 1998.

[116] Rockwell. Wireless sensing network (WSN). `http://wins.rsc.rockwell.com/`.

[117] Z. Roth and Y. Baram. Multidimensional density shaping by Sigmoids. *IEEE Transactions on Neural Networks*, 7(5):1291–1298, 1996.

[118] SensorView. `http://www.sensorview.com/`.

[119] S. Sampan. *Neural fuzzy techniques in vehicle acoustic signal classification*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, April 1997.

[120] A. M. Sayeed. UW-CSP for detection, localization, tracking and classification. `http://www.ece.wisc.edu/~sensit/presentations/csp\_apr\_15.ppt`, April 2001.

[121] `http://www.geo.mtu.edu/UPSeis/waves.html`.

[122] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[123] D. F. Shanno. Conditioning of Quasi-Newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.

[124] C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, pages 52–59, August 2001.

[125] D. Shepherd. Networked microsensors and the end of the world as we know it. *IEEE Technology and Society Magazine*, pages 16–22, Spring 2003.

[126] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks. In *Mobile Computing and Networking*, pages 272–286, July 2001.

[127] G. E. Sleefe and B. P. Engler. Experimental study of an advanced three-component borehole seismic receiver. In *The 61st Meeting of Society of Exploration Geophysicists*, pages 3–33, Houston, TX, 1991.

[128] G. E. Sleefe, M. D. Ladd, T. S. McDonald, and G. J. Elbring. Acoustic and seismic modalities for unattended ground sensors. In *Proceedings of SPIE: Unattended Ground Sensor Technologies and Applications*, volume 3713, pages 2–9, Orlando, FL, April 1999.

[129] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, October 2000.

[130] M. Southworth and D. Godso. Chem-bio sensor platform leverages pc/104. *COTS Journal*, pages 74–76, March 2003.

[131] M. Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environment. In *Proceedings on 7th International Conference on Mobile Computing and Networking (MobiCom 2001)*, pages 132–138, New York, 2001.

[132] L. G. Stotts. Steel rattler program overview. DIA Internal Report, 1995.

[133] G. Succi and T. K. Pedersen. Acoustic target tracking and target identification - recent results. In *Proceedings of SPIE: Unattended Ground Sensor Technologies and Applications*, volume 3713, pages 10–21, April 1999.

[134] T. Sundsted. An introduction to agents. *Java World*, June 1998.

[135] Crow Systems. PIR motion sensors - a basic overview for effective use. `http://crowsystems.hypermart.net/pir-faq.htm`.

[136] Home Security Burglar Alarm Systems. Motion sensor alarms. `http://www.home-security-burglar-alarms-systems.com`.

[137] Y. Tan and J. Wang. Nonlinear blind source separation using higher order statistics and a genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 5(6):600–612, 2001.

[138] Y. Tian. Target detection and classification using seismic signal processing in unattended ground sensor systems. Master's thesis, University of Tennessee, Knoxville, July 2001.

[139] Diuturnity testbed, USC Robotics Research Lab. `http://www-robotics.usc.edu/projects/robomote/testbed/`.

[140] X. Wang, H. Qi, and H Du. Distributed source number estimation for multiple target detection in sensor networks. In *IEEE Workshop on Statistical Signal Processing*, St.Louis, September 2003.

[141] X. Wang, H. Qi, and S. S. Iyengar. Collaborative multi-modality target classification in distributed sensor networks. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 285–290, Annapolis, MD, July 2002.

[142] M. Weiser. The computer for the 21st century. *Scientific American*, September 1991.

[143] P. D. Welch. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio Electroacoustics*, AU-15:70–73, June 1967.

[144] M. C. Wellman, N. Srour, and D. B. Hillis. Feature extraction and fusion of acoustic and seismic sensors for target identification. In *Proceedings of SPIE: Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors*, volume 3081, pages 139–145, April 1997.

[145] B. W. West, P. G. Flikkema, T. Sisk, and G. W. Koch. Wireless sensor networks for dense spatio-temporal monitoring of the environment: a case for integrated circuit, system, and network design. In *Proceedings of 2001 IEEE CAS Workshop on Wireless Communications and Networking*, August 2001.

[146] J. K. Wolford. Event identification from seismic/magnetic feature vectors - a comparative study. In *Proceedings of SPIE: Peace and Wartime Applications and Technical Issues for Unattended Ground Sensors*, volume 3081, pages 131–138, July 1997.

[147] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:418–435, 1992.

[148] Y. Xu and H. Qi. Performance evaluation of distributed computing paradigms in mobile ad hoc sensor networks. In *Accepted by International Conference on Parallel and Distributed Systems (ICPADS)*, Taiwan, December 2002.

[149] X. Yang, K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and C. A. Grimes. Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment. *Sensors*, 2(7):455–472, 2002.

[150] K. Yao, J. C. Chen, and R. E. Hudson. Maximum-likelihood acoustic source localization: experimental results. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 2949–2952, 2002.

[151] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.

Vita

Xiaoling Wang was born in Shenyang, P. R. China. After graduating in 1993 from Shenyang Railway High School, she attended Northeastern University in Shenyang exempted from the National Standard Entrance Exam, where she received both a Bachelor of Science degree in 1997 and a Master of Science degree in 2000 from the College of Information Science and Engineering. Her research area during this period is intelligent control and power system low-frequency protection. In the fall of 2000, Xiaoling enrolled into the doctoral program at the University of Tennessee in Electrical and Computer Engineering. At the same time, she joined the Advanced Imaging and Collaborative Information Processing group as a graduate research assistant where she completed her Doctor of Philosophy degree in 2004. She had been working on a 4-year project sponsored by Defense Advanced Research Projects Agency, titled “Sensor Information Technology Program”. Her major research areas are distributed signal/image processing, pattern recognition, and information fusion in sensor networks.