Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications

EDUARDO F. NAKAMURA

Analysis, Research and Technological Innovation Center – FUCAPI
Federal University of Minas Gerais – UFMG

ANTONIO A. F. LOUREIRO

Federal University of Minas Gerais – UFMG

and

ALEJANDRO C. FRERY

Federal University of Alagoas – UFAL

Wireless sensor networks produce a large amount of data that needs to be processed, delivered, and assessed according to the application objectives. The way these data are manipulated by the sensor nodes is a fundamental issue. Information fusion arises as a response to process data gathered by sensor nodes and benefits from their processing capability. By exploiting the synergy among the available data, information fusion techniques can reduce the amount of data traffic, filter noisy measurements, and make predictions and inferences about a monitored entity. In this work, we survey the current state-of-the-art of information fusion by presenting the known methods, algorithms, architectures, and models of information fusion, and discuss their applicability in the context of wireless sensor networks.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.9 [**Artificial Intelligence**]: Robotics—*Sensors*; I.5.1 [**Pattern Recognition**]: Models; I.5.2 [**Pattern Recognition**]: Methodology; I.5.4 [**Pattern Recognition**]: Applications; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks, Network topology*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.3 [**Special-Purpose and Application-Based Systems**]:—*Real-time and embedded systems, Signal processing systems*; H.4.2 [**Information Systems Applications**]: types of Systems—*Decision support*

General Terms: Algorithms, Design, Measurement

Additional Key Words and Phrases: Information fusion, data aggregation, data fusion, wireless sensor networks, architectures and models

Authors' addresses: E. F. Nakamura, Analysis, Research and Technological Innovation Center (FUCAPI), Av. Gov. Danilo de Matos Areosa, 381, DI, 69075-351, Manaus, AM, Brazil; email: eduardo.nakamura@fucapi.br; A. A. F. Loureiro, Department of Computer Science, Federal University of Minas Gerais, Av. Antonio Carlos, 6627, 31270-010, Belo Horizonte, MG, Brazil; email: loureiro@dcc.ufmg.br; A. C. Frery, Department of Information Technology, Federal University of Alagoas; Campus A. C. Simoes, BR 104 Norte km 97, Tabuleiro do Martins, 57072-970, Maceio, AL, Brazil; email: frery@tci.ufal.br.

ACM Reference Format:

Nakamura, E. F., Loureiro, A. A. F., and Frery, A. C. 2007. Information fusion for wireless sensor networks: methods, models, and classifications. *ACM Comput. Surv*. 39, 3, Article 9 (August 2007), 55 pages DOI = 10.1145/1267070.1267073 http://doi.acm.org/10.1145/1267070.1267073

1. INTRODUCTION

A Wireless Sensor Network (WSN) [Pottie and Kaiser 2000; Akyildiz et al. 2002] is a special type of *ad hoc* network composed of a large number of nodes equipped with different sensor devices. This network is supported by technological advances in low-power wireless communications along with silicon integration of various functionalities such as sensing, communication, and processing. WSNs are emerging as an important computer class based on a new computing platform and networking structure that will enable novel applications that are related to different areas such as environmental monitoring, industrial and manufacturing automation, health-care, and military. Commonly, wireless sensor networks have strong constraints regarding power resources and computational capacity.

A WSN may be designed with different objectives. It may be designed to gather and process data from the environment in order to have a better understanding of the behavior of the monitored entity. It may also be designed to monitor an environment for the occurrence of a set of possible events, so that the proper action may be taken whenever necessary. A fundamental issue in WSNs is the way the collected data is processed. In this context, information fusion arises as a discipline that is concerned with how data gathered by sensors can be processed to increase the relevance of such a mass of data. In a nutshell, information fusion can be defined as the combination of multiple sources to obtain improved information (cheaper, greater quality, or greater relevance).

Information fusion is commonly used in detection and classification tasks in different application domains, such as robotics and military applications [Brooks and Iyengar 1998]. Lately, these mechanisms have been used in new applications such as intrusion detection [Bass 2000] and Denial of Service (DoS) detection [Siaterlis and Maglaris 2004]. Within the WSN domain, simple aggregation techniques (e.g., *maximum*, *minimum*, and *average*) have been used to reduce the overall data traffic to save energy [Intanagonwiwat et al. 2000; Krishnamachari et al. 2002; Madden et al. 2002]. Additionally, information fusion techniques have been applied to WSNs to improve location estimates of sensor nodes [Savvides et al. 2003], detect routing failures [Nakamura et al. 2005b], and collect link statistics for routing protocols [Woo et al. 2003].

Given the importance of information fusion for WSNs, this work surveys the state-of-the-art related to information fusion and how it has been used in WSNs and sensor-based systems in general. This background is presented in the following structure. Section 2 presents the common terminology used to describe information fusion. Common classifications are discussed in Section 3. The main methods are presented in Section 4. Section 5 presents the current architectures for information fusion and discusses their limitations and applicability to WSNs. Section 7 presents our final remarks on how current information fusion resources can be applied to the context of WSNs.

2. FUNDAMENTALS

Several different terms (e.g. data fusion, sensor fusion, and information fusion) have been used to describe aspects of the fusion subject (including theories, processes, systems, frameworks, tools, and methods). Consequently, there is a terminology confusion.

This section discusses common terms and factors that motivate and encourage the practical use of information fusion in WSNs.

2.1. The Name of the Game

The terminology related to systems, architectures, applications, methods, and theories about the fusion of data from multiple sources is not unified. Different terms have been adopted, usually associated with specific aspects that characterize the fusion. For example, Sensor/Multisensor Fusion is commonly used to specify that sensors provide the data being fused. Despite the philosophical issues about the difference between data and information, the terms Data Fusion and Information Fusion are usually accepted as overall terms.

Many definitions of data fusion have been provided through the years, most of them derived from military and remote sensing fields. In 1991, the data fusion work group of the Joint Directors of Laboratories (JDL) organized an effort to define a lexicon [U.S. Department of Defense 1991] with some terms of reference for data fusion. They define data fusion as a "multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from multiple sources." Klein [1993] generalizes this definition, stating that data can be provided by a single source or by multiple sources. Both definitions are general and can be applied in different fields, including remote sensing. Although they suggest the combination of data without specifying its importance nor its objective, the JDL data fusion model provided by the U.S. Department of Defense [1991] deals with quality improvement, which will be further discussed in Section 5.

Hall and Llinas [1997] define data fusion as "the combination of data from multiple sensors, and related information provided by associated databases, to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone." Here, data fusion is performed with an objective: accuracy improvement. However this definition is restricted to data provided by sensors; it does not foresee the use of data from a single source.

Claiming that all previous definitions are focused on methods, means, and sensors, Wald [1999] changes the focus to the framework used to fuse data. Wald states that "data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application." In addition, Wald considers data taken from the same source at different instants as distinct sources. The word "quality" is a loose term intentionally adopted to denote that the fused data is somehow more appropriate to the application than the original data. In particular for WSNs, data can be fused with at least two objectives: accuracy-improvement and energy-saving.

Although Wald's definition and terminology are well accepted by the Geoscience and Remote Sensing Society [2004], and officially adopted by the Data Fusion Server [2004], the term Multisensor Fusion has been used with the same meaning by other authors, such as Hall [1992], and Waltz and Llinas [1990].

Multisensor Integration is another term used in robotics/computer vision [Luo and Kay 1995] and industrial automation [Brokmann et al. 2001]. According to Luo et al. [2002], multisensor integration "is the synergistic use of information provided by multiple sensory devices to assist in the accomplishment of a task by a system; and multisensor fusion deals with the combination of different sources of sensory information into one representational format during any stage in the integration process." Multisensor integration is a broader term than multisensor fusion. It makes explicit how the fused data is used by the whole system to interact with the environment.

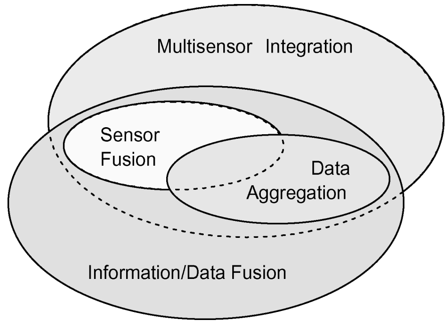

Fig. 1. The relationship among the fusion terms: multisensor/sensor fusion, multisensor integration, data aggregation, data fusion, and information fusion.

However, it might suggest that only sensory data is used in the fusion and integration processes.

This confusion of terms is highlighted by Dasarathy [1997] who adopted the term Information Fusion [Dasarathy 2001] stating that "in the context of its usage in the society, it encompasses the theory, techniques and tools created and applied to exploit the synergy in the information acquired from multiple sources (sensor, databases, information gathered by humans, etc.) in such a way that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness, etc.) than would be possible if any of these sources were used individually without such synergy exploitation." Possibly, this is the broadest definition embracing any type of source, knowledge, and resource used to fuse different pieces of information. The term Information Fusion and Dasarathy's definition are also adopted by the International Society of Information Fusion [2004]. Kokar et al. [1999] also use this term in a framework of formal logic and category theory where the structures representing the meaning of information (theories and models) are actually fused, while data is just processed and filtered through such structures.

The term Data Aggregation has become popular in the wireless sensor network community as a synonym for information fusion [Kalpakis et al. 2003; van Renesse 2003]. According to Cohen et al. [2001], "data aggregation comprises the collection of raw data from pervasive data sources, the flexible, programmable composition of the raw data into less voluminous refined data, and the timely delivery of the refined data to data consumers." By using 'refined data', accuracy improvement is suggested. However, as van Renesse [2003] defines, "aggregation is the ability to summarize," which means that the amount of data is reduced. For instance, by means of summarization functions, such as *maximum* and *average*, the volume of data being manipulated is reduced. However, for applications that require original and accurate measurements, such a summarization may represent an accuracy loss [Boulis et al. 2003a]. In fact, although many applications might be interested only in summarized data, we cannot always assert whether or not the summarized data is more accurate than the original data-set. For this reason, the use of data aggregation as an overall term should be avoided because it also refers to one instance of information fusion: summarization.

Figure 1 depicts the relationship among the concepts of multisensor/sensor fusion, multisensor integration, data aggregation, data fusion, and information fusion. Here, we understand that both terms, data fusion and information fusion, can be used with the same meaning. Multisensor/sensor fusion is the subset that operates with sensory sources. Data aggregation defines another subset of information fusion that aims to reduce the data volume (typically, summarization), which can manipulate any type of data/information, including sensory data. On the other hand, multisensor integration is a slightly different term in the sense that it applies information fusion to make inferences using sensory devices and associated information (e.g., from database systems) to interact with the environment. Thus, multisensor/sensor fusion is fully contained in the intersection of multisensor integration and information/data fusion.

Here, we chose to use information fusion as the overall term so that sensor and multisensor fusion can be considered as the subset of information fusion that handles data acquired by sensory devices. However, as data fusion is also accepted as an overall term, we reinforce Elmenreich's recommendation [Elmenreich 2002], which states that fusion of raw (or low level) data should be explicitly referred to as *raw data fusion* or *low level data fusion* to avoid confusion with the *data fusion* term used by the Geoscience and Remote Sensing Society [2004].

2.2. The Whys and Wherefores of Information Fusion

WSNs are intended to be deployed in environments where sensors can be exposed to conditions that might interfere with their measurements. Such conditions include strong variations of temperature and pressure, electromagnetic noise, and radiation. Therefore, sensors' measurements may be imprecise (or even useless) in such scenarios. Even when environmental conditions are ideal, sensors may not provide perfect measurements. Essentially, a sensor is a measurement device, and an imprecision value is usually associated with its observation. Such imprecision represents the imperfections of the technology and methods used to measure a physical phenomenon or property.

Failures are not an exception in WSNs. For instance, consider a WSN that monitors a forest to detect an event, such as fire or the presence of an animal. Sensor nodes can be destroyed by fire, animals, or even human beings; they might present manufacturing problems; and they might stop working due to a lack of energy. Each node that becomes inoperable might compromise the overall perception and/or the communication capability of the network. Here, perception capability is equivalent to the exposure concept [Meguerdichian et al. 2001a; Megerian et al. 2002].

Both spatial and temporal coverage also pose limitations to WSNs. The sensing capability of a node is restricted to a limited region. For example, a thermometer in a room reports the temperature near the device but it might not fairly represent the overall temperature inside the room. Spatial coverage in WSNs [Meguerdichian et al. 2001b] has been explored in different scenarios, such as target tracking [Chakrabarty et al. 2002], node scheduling [Tian and Georganas 2002], and sensor placement [Dhillon et al. 2002]. Temporal coverage can be understood as the ability to fulfill the network purpose during its lifetime. For instance, in a WSN for event detection, temporal coverage aims at assuring that no relevant event will be missed because there was no sensor perceiving the region at the specific time the event occurred. Thus, temporal coverage depends on the sensor's sampling rate, communication delays, and the node's duty cycle (time when it is awake or asleep).

To overcome sensor failures, technological limitations, spatial, and temporal coverage problems, three properties must be ensured: *cooperation*, *redundancy*, and *complementarity* [Durrant-Whyte 1988; Luo et al. 2002]. Usually, a region of interest can only be fully covered by the use of several sensor nodes, each cooperating with a partial view of the scene; information fusion can be used to compose the complete view from the pieces provided by each node. Redundancy makes the WSN less vulnerable to failure of a single node, and overlapping measurements can be fused to obtain more accurate data; Rao [2001] shows how information fusion can perform at least as well as the best sensor. Complementarity can be achieved by using sensors that perceive different properties of the environment; information fusion can be used to combine complementary data so the resultant data allows inferences that might be not possible to be obtained from the individual measurements (e.g., angle and distance of an imminent threat can be fused to obtain its position).

Due to redundancy and cooperation properties, WSNs are often composed of a large number of sensor nodes posing a scalability challenge caused by potential collisions and transmissions of redundant data. Regarding the energy restrictions, communication should be reduced to increase the lifetime of the sensor nodes. Thus, information fusion is also important to reduce the overall communication load in the network, by avoiding the transmission of redundant messages. In addition, any task in the network that handles signals or needs to make inferences, can potentially use information fusion.

2.3. Some Limitations

Information fusion should be considered a critical step in designing a wireless sensor network. The reason is that information fusion can be used to extend the network lifetime and is commonly used to fulfill application objectives, such as target tracking, event detection, and decision making. Hence, blundering information fusion may result in waste of resources and misleading assessments. Therefore, we must be aware of possible limitations of information fusion to avoid blundering situations.

Because of resource rationalization needs of WSNs, data processing is commonly implemented as in-network algorithms [Akyildiz et al. 2002; Intanagonwiwat et al. 2003; Madden et al. 2005]. Hence, whenever possible, information fusion should be performed in a distributed (in-network) fashion to extend the network lifetime. Nonetheless, we must be aware of the limitations of distributed implementations of information fusion.

In the early 1980s, Tenney and Sandell [1981] argued that, regarding the communication load, a centralized fusion system may outperform a distributed one. The reason is that centralized fusion has a global knowledge in the sense that all measured data is available, whereas distributed fusion is incremental and localized since it fuses measurements provided by a set of neighbor nodes and the result might be further fused by intermediate nodes until a sink node is reached. Such a drawback of decentralized fusion might often be present in WSNs wherein, due to resource limitations, distributed and localized algorithms are preferable to centralized ones. In addition, the lossy nature of wireless communication challenges information fusion because losses mean that input data may not be completely available.

Another issue regarding information fusion is that, intuitively, one might believe that in fusion processes, the more data the better, since the additional data should add knowledge (e.g., to support decisions or filter embedded noise). However, as Dasarathy [2000] shows, when the amount of additional incorrect data is greater than the amount of additional correct data, the overall performance of the fusion process can be reduced.

3. CLASSIFYING INFORMATION FUSION

Information fusion can be categorized based on several aspects. Relationships among the input data may be used to segregate information fusion into classes (e.g., cooperative, redundant, and complementary data). Also, the abstraction level of the manipulated data during the fusion process (measurement, signal, feature, decision) can be used to distinguish among fusion processes. Another common classification consists in making explicit the abstraction level of the input and output of a fusion process. These common classifications of information fusion are explored in this section.

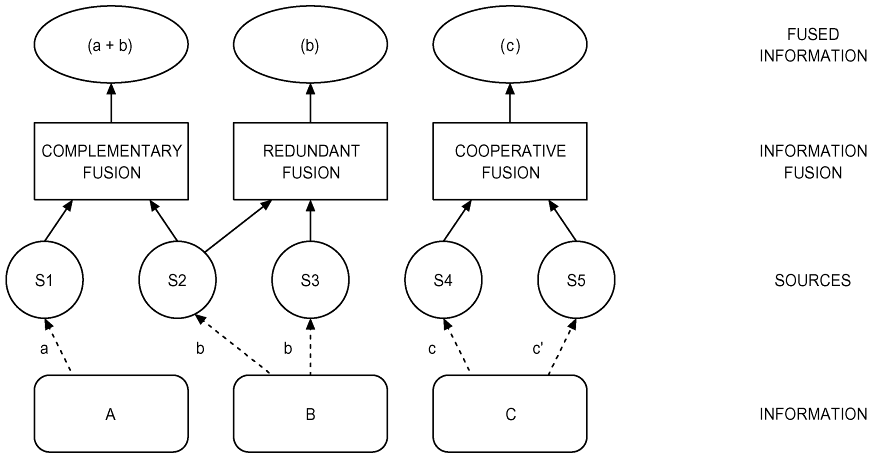

Fig. 2. Types of information fusion based on the relationship among the sources, figure adapted from Elmenreich [2002].

3.1. Classification Based on Relationship Among the Sources

According to the relationship among the sources, information fusion can be classified as complementary, redundant, or cooperative [Durrant-Whyte 1988]. Thus, according to the relationship among sources, information fusion can be:

Complementary. When information provided by the sources represents different portions of a broader scene, information fusion can be applied to obtain a piece of information that is more complete (broader). In Figure 2, sources S1 and S2 provide different pieces of information, a and b, respectively, that are fused to achieve a broader information, denoted by $(a+b)$, composed of nonredundant pieces a and b that refer to different parts of the environment (e.g., temperature of west and east sides of the monitored area).

Redundant. If two or more independent sources provide the same piece of information, these pieces can be fused to increase the associated confidence. Sources S2 and S3 in Figure 2 provide the same information, b, which is fused to obtain more accurate information, (b).

Cooperative. Two independent sources are cooperative when the information provided by them is fused into new information (usually more complex than the original data) that, from the application perspective, better represents the reality. Sources S4 and S5, in Figure 2, provide different information, c and c', that are fused into (c), which better describes the scene compared to c and c' individually.

Complementary fusion searches for completeness by compounding new information from different pieces. An example of complementary fusion consists in fusing data from sensor nodes (e.g., a sample from the sensor field) into a feature map that describes the whole sensor field [Zhao et al. 2002b; Willett et al. 2004; Nowak et al. 2004; Singh et al. 2006], hence broader information.

Redundant fusion might be used to increase the reliability, accuracy, and confidence of the information. In WSNs, redundant fusion can provide high quality information and prevent sensor nodes from transmitting redundant information. Typical examples of redundant fusion are filters discussed in Section 4.2 whose estimates are improved when additional redundant information is available.

A classical example of cooperative fusion is the computation of a target location based on angle and distance information. Cooperative fusion should be carefully applied since the resultant data is subject to the inaccuracies and imperfections of all participating sources [Brooks and Iyengar 1998].

3.2. Classification Based on Levels of Abstraction

Luo et al. [2002] use four levels of abstraction to classify information fusion: signal, pixel, feature, and symbol. Signal level fusion deals with single or multidimensional signals from sensors. It can be used in real-time applications or as an intermediate step for further fusions. Pixel level fusion operates on images and can be used to enhance image-processing tasks. Feature level fusion deals with features or attributes extracted from signals or images, such as shape and speed. In symbol level fusion, information is a symbol that represents a decision, and it is also referred to as decision level. Typically, the feature and symbol fusions are used in object recognition tasks. Such a classification presents some drawbacks and is not suitable for all information fusion applications. First, both signals and images are considered raw data usually provided by sensors, so they might be included in the same class. Second, raw data may not be only from sensors, since information fusion systems might also fuse data provided by databases or human interaction. Third, it suggests that a fusion process cannot deal with all levels simultaneously.

In fact, information fusion deals with three levels of data abstraction: measurement, feature, and decision [Dasarathy 1997; Iyengar et al. 2001]. Thus, according to the abstraction level of the manipulated data, information fusion can be classified into four categories:

Low-Level Fusion. Also referred to as *signal (measurement) level fusion*. Raw data are provided as inputs, combined into new piece of data that is more accurate (reduced noise) than the individual inputs. Polastre et al. [2004] provide an example of low-level fusion by applying a moving average filter (we discuss such filters in Section 4.2.4) to estimate ambient noise and determine whether or not the communication channel is clear.

Medium-Level Fusion. Attributes or features of an entity (e.g., shape, texture, position) are fused to obtain a feature map that may be used for other tasks (e.g., segmentation or detection of an object). This type of fusion is also known as *feature/attribute level fusion*. Examples of this type of information fusion include estimation of fields or feature maps [Nowak et al. 2004; Singh et al. 2006] and energy maps [Zhao et al. 2002b; Mini et al. 2004] (see Section 4.3 for a feature map description).

High-Level Fusion. Also known as *symbol* or *decision level fusion*. It takes decisions or symbolic representations as input and combines them to obtain a more confident and/or a global decision. An example of high-level fusion is the Bayesian approach for binary event detection proposed by Krishnamachari and Iyengar [2004], which detects and corrects measurement faults.

Multilevel Fusion. When the fusion process encompasses data of different abstraction levels—when both input and output of fusion can be of any level (e.g., a measurement is fused with a feature to provide a decision)—multilevel fusion takes place. Nakamura et al. [2005b] provide an example of multilevel fusion by applying the Dempster-Shafer (see Section 4.1.2) theory to decide about node failures based on traffic decay features.

Although the first three levels of fusion are specified by Iyengar et al. [2001], they do not specify Multilevel Fusion. Typically, only the first three categories of fusion (low, medium, and high level) are considered, usually with the terms *pixel/measurement*, *feature*, and *decision fusion* [Pohl and van Genderen 1998]. However, such a categorization does not foresee the fusion of information of different levels of abstraction at the same time. For example, the fusion of a signal or an image with a feature resulting in a decision [Dasarathy 1997; Wald 1999].

3.3. Classification Based on Input and Output

Another well-known classification that considers the abstraction level is provided by Dasarathy [1997], in which information fusion processes are categorized based on the abstraction level of the input and output information. Dasarathy identifies five categories:

Data In–Data Out (DAI-DAO). In this class, information fusion deals with raw data and the result is also raw data, possibly more accurate or reliable.

Data In–Feature Out (DAI-FEO). Information fusion uses raw data from sources to extract features or attributes that describe an entity. Here, "entity" means any object, situation, or world abstraction.

Feature In–Feature Out (FEI-FEO). FEI-FEO fusion works on a set of features to improve/refine a feature, or extract new ones.

Feature In–Decision Out (FEI-DEO). In this class, information fusion takes a set of features of an entity generating a symbolic representation or a decision.

Decision In–Decision Out (DEI-DEO) Decisions can be fused in order to obtain new decisions or give emphasis on previous ones.

Compared to the classification presented in Section 3.2, this classification can be seen as an extension of the previous one but with a finer granularity, where DAI-DAO corresponds to Low Level Fusion, FEI-FEO to Medium Level Fusion, DEI-DEO to High Level Fusion, and DAI-FEO and FEI-DEO are included in Multilevel Fusion. Contextualizing the examples in Section 3.2, Polastre et al. [2004] use DAI-DAO fusion for ambient noise estimation through a moving average filter; Singh et al. [2006] use FEI-FEO fusion for building feature maps that geographically describe a sensed parameter such as temperature; Luo et al. [2006] use DEI-DEO fusion for binary event detection by fusing several single detections (sensor reports) to decide about an actual event detection; and Nakamura et al. [2005b] apply FEI-DEO fusion when they fuse features describing the traffic decay to infer about node failures.

The main contribution of Dasarathy's classification is that it specifies the abstraction level of both input and output of a fusion process, avoiding possible ambiguities. However, it does not allow in the same process, the fusion, for instance, of features and signals to refine a given feature or provide a decision.

4. METHODS, TECHNIQUES, AND ALGORITHMS

Methods, techniques, and algorithms used to fuse data can be classified based on several criteria, such as the data abstraction level, purpose, parameters, type of data, and mathematical foundation. The classification presented in this section is based on the method's purpose. According to this criterion, information fusion can be performed with different objectives such as inference, estimation, classification, feature maps, abstract sensors, aggregation, and compression.

4.1. Inference

Inference methods are often applied in decision fusion. In this case, a decision is taken based on the knowledge of the perceived situation. Here, inference refers to the transition from one likely true proposition to another, whose truth is believed to result from the previous one. Classical inference methods are based on Bayesian inference and Dempster-Shafer Belief Accumulation theory.

4.1.1. Bayesian Inference. Information fusion based on Bayesian Inference offers a formalism to combine evidence according to rules of probability theory. The uncertainty is represented in terms of conditional probabilities describing the belief, and it can assume values in the [0, 1] interval, where 0 is absolute disbelief and 1 is absolute belief. Bayesian inference is based on the rather old Bayes' rule [Bayes 1763], which states that:

$$\Pr(Y \mid X) = \frac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)}, \tag{1}$$

where the posterior probability $\Pr(Y \mid X)$ represents the belief of hypothesis Y given the information X. This probability is obtained by multiplying $\Pr(Y)$, the prior probability of the hypothesis Y, by $\Pr(X \mid Y)$, the probability of receiving X, given that Y is true; $\Pr(X)$ can be treated as a normalizing constant. The main issue regarding Bayesian Inference is that the probabilities $\Pr(X)$ and $\Pr(X \mid Y)$ have to be estimated or guessed beforehand since they are unknown.

Pan et al. [1998] propose the use of neural networks to estimate conditional probabilities to feed a Bayesian Inference module for decision-making. Sam et al. [2001] use Bayesian Inference to decide whether or not a system's voltage is stable by fusing three stability indicators of a small power system. Coué et al. [2002] use Bayesian programming, a general approach based on an implementation of Bayesian theory, to fuse data from different sensors (e.g., laser, radar, and video) to achieve better accuracy and robustness of the information required for high-level driving assistance. Typical usage for Bayesian Inference includes robotic map building [Moshiri et al. 2002] and classification tasks [Tsymbal et al. 2003].

Within the WSNs domain, Bayesian Inference has been used to solve the localization problem. Particularly, Sichitiu and Ramadurai [2004] use Bayesian Inference to process information from a mobile beacon and determine the most likely geographical location (region) of each node, instead of finding a unique point for each node location. Biswas et al. [2004] model the sensor network as a Bayesian network and use Markov Chain Monte Carlo sampling [Gilks et al. 1996] to infer whether or not a friendly agent is surrounded by enemy agents. A breakthrough work in event detection for wireless sensor networks is proposed by Krishnamachari and Iyengar [2004] who explicitly consider measurement faults and develop a distributed and localized Bayesian algorithm for detecting and correcting such faults. This work is further extended by Luo et al. [2006] who consider both measurement errors and sensor faults in the detection task. The BARD approach [Stann and Heidemann 2005] uses Bayesian Inference to reduce the communication costs related to resource and route discovery by limiting the associated communication to the nodes that are most likely to connect a source to a sink node. The *infer* algorithm [Hartl and Li 2005] is a distributed solution that uses Bayesian Inference to determine the missing data from the nodes that are not active (sleep mode) during a sensing epoch.

4.1.2. Dempster-Shafer Inference. Dempster-Shafer Inference is based on the Dempster-Shafer Belief Accumulation (also referred to as Theory of Evidence or Dempster-Shafer Evidential Reasoning), which is a mathematical theory introduced by Dempster [1968] and Shafer [1976] that generalizes the Bayesian theory. It deals with beliefs or mass functions just as Bayes' rule does with probabilities. The Dempster-Shafer theory provides a formalism that can be used for incomplete knowledge representation, belief updates, and evidence combination [Provan 1992].

A fundamental concept in a Dempster-Shafer reasoning system is the *frame of discernment*, which is defined as follows. Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ be the set of all possible states that describe the system, such that Θ is exhaustive and mutually exclusive in the sense that the system is certainly in one, and only one, state $\theta_i \in \Theta$, where $1 \leq i \leq N$. We call Θ the frame of discernment because its elements are used to discern the actual system states.

The elements of the power set 2^{Θ} are called hypotheses. In the Dempster-Shafer theory, based on evidence E, a probability is assigned to every hypothesis $H \in 2^{\Theta}$ according to a *basic probability assignment* (bpa), or mass function, $m\colon 2^{\Theta} \rightarrow [0, 1]$ that satisfies:

$$m(\emptyset) = 0 \tag{2}$$
$$m(H) \geq 0, \forall\, H \in 2^{\Theta} \tag{3}$$
$$\sum_{H \in 2^{\Theta}} m(H) = 1. \tag{4}$$

To express the overall belief in a hypothesis H, Dempster-Shafer defines the belief function $bel\colon 2^{\Theta} \rightarrow [0, 1]$ over Θ as:

$$bel(H) = \sum_{A \subseteq H} m(A), \tag{5}$$

where $bel(\emptyset) = 0$, and $bel(\Theta) = 1$.

The degree of doubt in H can be intuitively expressed in terms of the belief function $bel\colon 2^{\Theta} \rightarrow [0, 1]$ as:

$$dou(H) = bel(\neg H) = \sum_{A \subseteq \neg H} m(A). \tag{6}$$

To express the plausibility of each hypothesis, the function $pl\colon 2^{\Theta} \rightarrow [0, 1]$ over Θ is defined as:

$$pl(H) = 1 - dou(H) = \sum_{A \cap H = \emptyset} m(A). \tag{7}$$

The plausibility intuitively states that the less the doubt in hypothesis H, the more plausible. In this context, the confidence interval $[bel(H), pl(H)]$ defines the true belief of the hypothesis H.

To combine the effects of two bpa's m_1 and m_2, the Dempster-Shafer theory defines a combination rule, $m_1 \oplus m_2$, which is given by:

$$m_1 \oplus m_2(\emptyset) = 0, \tag{8}$$
$$m_1 \oplus m_2(H) = \frac{\sum_{X \cap Y = H} m_1(X) m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)}. \tag{9}$$

According to Luo and Kay [1992], the use of the Dempster-Shafer theory for information fusion of sensory data was introduced in 1981 by Garvey et al. [1981]. The Dempster-Shafer theory is more flexible than Bayesian Inference for it allows each source to contribute information with different levels of detail. To illustrate this assertion, let us suppose we have two sensors, A and B, able to distinguish the roar of male from female felines; and we also have a third sensor, C, that distinguishes a cheetah roar from a lion roar. In this scenario, we can use the Dempster-Shafer theory to fuse data from the three sensors to detect male/female lions and male/female cheetahs, while such an inference would be more difficult with a Bayesian method. The reason is that, in contrast to the Bayesian Inference, the Dempster-Shafer theory allows us to fuse data provided by different types of sensors. Furthermore, in the Dempster-Shafer inference we do not need to assign a priori probabilities to unknown propositions. Instead, probabilities are assigned only when the supporting information is available.

Choosing between the Bayesian Inference and the Theory of Evidence is not a trivial task because, among other things, there is a tradeoff between Bayesian accuracy and Dempster-Shafer flexibility [Bracio et al. 1997]. Comparisons between these two inference methods are provided by Buede [1988] and Cheng and Kashyap [1988].

Pinto et al. [2004] discuss in-network implementations of Dempster-Shafer and the Bayesian inference in such a way that event detection and data routing are unified into a single algorithm. By using a WSN composed of Unmanned Aerial Vehicles (UAVs) as sensor nodes, Yu et al. [2004] use the Dempster-Shafer inference to build dynamic operational pictures of battlefields for situation assessment. However, the particular challenges of in-network fusion in such a mobile network are not evaluated. In the Data Service Middleware (DSWare) for WSNs designed by Li et al. [2004], every decision is associated with a confidence value that is computed by a prespecified confidence function based on the belief and plausibility functions of the Dempster-Shafer theory. In a different application, Nakamura et al. [2005b] propose the Topology Rebuilding Algorithm (TRA) as an improvement to tree-based routing algorithms. The TRA algorithm analyzes data traffic and uses the Dempster-Shafer inference to detect routing failures, and trigger a topology reconstruction (route re-discovery) only when necessary.

4.1.3. Fuzzy Logic. Fuzzy logic generalizes probability [Banon 1981] and therefore is able to deal with approximate reasoning [Novák et al. 1999] to draw (possibly imprecise) conclusions from imprecise premises. Each quantitative input is fuzzyfied by a membership function. The fuzzy rules of an inference system produce fuzzy outputs which, in turn, are defuzzyfied by a set of output rules. This framework has been successfully used in real world situations that defy exact modelling, from rice cookers[1] to complex control systems [Lee 1990].

Cui et al. [2004] study the problem of controlling the position of sensors for localizing hazardous contaminant sources. They propose a fuzzy logic position control algorithm able to cope with the incomplete, uncertain, and approximate information the sensor gathers. The purpose of the algorithm is manyfold, namely, exploring the whole area, keeping connectivity and finding the emission source. Aiming at optimizing mobile sensor deployment, Shu and Liang [2005] update the position of each node using a fuzzy optimization algorithm. This technique fuzzyfies the number of neighbors of each sensor and the average distance among them in order to derive an updating rule.

An intelligent sensor network and fuzzy logic control are used by Chan Yet and Qidwai [2005] to develop an autonomous navigational robotic vehicle with obstacle avoidance capability. The navigation is guided by two controllers: one for detecting potholes and another for avoiding obstacles. The input to each controller is the stereoscopic information gathered by ultrasonic sensors, and the fuzzyficaton is performed using training data obtained beforehand. These two subsystems feed the main controller, which decides the best path to follow.

Gupta et al. [2005] and Halgamuge et al. [2003] use fuzzy reasoning for deciding the best cluster-heads in a WSN. The former use three features to guide the choice: node concentration, energy level, and centrality. After fuzzyfication, these features are turned into linguistic variables and a rule is obtained. The technique proves to be better than the stochastic procedure proposed by Heinzelman et al. [2000]. The latter use energy measures and a fuzzy clustering algorithm, and their results are better than those of a substractive clustering technique [Bezdek 1981].

Regarding the design of Medium Access Control (MAC) protocols, Wallace et al. [2005] propose a two-stage fuzzy-based control aiming at prolonging the network lifetime. The

[1] `http://en.wikipedia.org/wiki/Rice_cooker`

inputs for the first stage are, for each node, size of the current transmit queue, remaining battery level, and collision of previous packages. The second stage gives priority to access the medium to nodes with a high transmit queue using the three previous inputs as well. The authors show that their sleeping duty cycles extend the network lifetime with respect to a fixed cycle strategy. With the same purpose, Liang and Ren [2005b] propose a MAC protocol with a fuzzy logic rescheduling scheme that improves existing energy-efficient protocols, among other advantages. Their input variables are the ratios of nodes (i) with overflowed buffer, (ii) with high failing transmission rate, and (iii) experiencing unsuccessful transmission.

Efficient routing is another area where fuzzy logic is used, aiming at the optimization of energy usage in WSNs. Yusuf and Haider [2005] assume a cluster-based architecture and study gateway centralized intercluster routing. They use transmission energy, remaining energy, rate of energy consumption, queue size, distance from the gateway, and current status as input variables; the fuzzy output is the cost. Liang and Ren [2005a] use battery capacity, mobility, and distance to the destination as variables for a fuzzy system that improves network lifetime by deciding the possibility of each node being included in the path. Srinivasan et al. [2006] use a fuzzy system to infer the ability of each node to transmit data using its battery power and the type of data being forwarded; and during route discovery, the output of the fuzzy logic controller is used to decide whether or not to forward a packet.

4.1.4. Neural Networks. According to Bonissone [1997], neural networks were originated in the early 1960s with Rosenblatt [1959] and Widrow and Hoff [1960]. They are structures that implement supervised learning mechanisms that starting from examples, are able to generalize. There are also unsupervised neural networks such as the Kohonen maps [Kohonen 1997]. Neural Networks represent an alternative to Bayesian and Dempster-Shafer theories, being used by classification and recognition tasks in the information fusion domain.

A key feature of neural networks is the ability of learning from examples of input/output pairs in a supervised fashion. For that reason, neural networks can be used in learning systems while fuzzy logic [Zadeh 1994] is used to control its learning rate [Bonissone 1997].

Neural networks have been applied to information fusion mainly for *Automatic Target Recognition* (ATR) using multiple complementary sensors [Luo and Kay 1992; Roth 1990; Filippidis et al. 2000]. The reason is that neural networks provide highly parallel means of processing yielding, thus robustness in the face of different issues such as noise [Castelaz 1988]. Baran [1989] proposes an information fusion approach for ATR that uses a neural network acting as an associative memory that guides the pattern-matching process for target recognition. Cain et al. [1989] use neural networks to classify targets based on information acquired from a multispectral infrared sensor and an ultraviolet laser radar.

Neural networks for information fusion can also be found in other applications besides ATR. Lewis and Powers [2002] use neural networks to fuse audio-visual information for audio-visual speech recognition. Cimander et al. [2002] use a two-stage fusion method that operates on signals from bioreactors (e.g., temperature, pH, and oxygen) to control the yogurt fermentation process. Yiyao et al. [2001] propose a fusion scheme named Knowledge-Based Neural Network Fusion (KBNNF) to fuse edge maps from multispectral sensor images acquired from radars, optical sensors, and infrared sensors.

4.1.5. Abductive Reasoning. Abduction, or inference to the best explanation, is a reasoning method in which we choose the hypothesis that would, if true, best explain an observed evidence [Peirce 1955]. In other words, once a fact is observed, abduction derives the most likely explanation for that fact. Thus, given a rule like $a \rightarrow b$ (a entails b), abduction and deduction differ in the direction in which the rule is used for inference. In *Deduction*, once the fact a is observed, we derive b as a consequence of a, whereas in *Abduction*, once the consequence b is observed, we derive a as an explanation of b. In its simplest case, abduction takes the form:

The fact b is observed; a, if true, explains b; No other hypothesis explains b better than a; Therefore, a is probably correct.

In the context of probabilistic reasoning, abductive inference corresponds to finding the maximum a posteriori probability state of the system's variables, given some observed variables [de Campos et al. 2002]. However, abduction is actually a reasoning pattern rather than an information fusion method. Hence, different inference methods can be used, such as Neural Networks [Abdelbar et al. 2003] and Fuzzy Logic [Aguero and Vargas 2005].

Since abduction searches for explanations, it is naturally applicable to diagnosis problems [Sidhu et al. 1997; Aguero and Vargas 2005], but has also been applied in crime investigation [Keppens et al. 2005], computer vision [Kumar and Desai 1996], and general machine learning problems [Mooney 2000]. Even though it has not been formally used in WSNs, abduction has great potential for different applications such as fault diagnosis, event detection and explanation, and environmental phenomena assessment.

4.1.6. Semantic Information Fusion. Semantic Information Fusion is essentially an in-network inference process in which raw sensor data is processed so that nodes exchange only the resulting semantic interpretations. The semantic abstraction allows a WSN to optimize its resource utilization when collecting, storing, and processing data. Semantic Information Fusion usually comprises two phases: knowledge base construction and pattern matching (inference). The first phase (usually off-line) aggregates the most appropriate knowledge abstractions into semantic information, which is then used in the second phase (on-line), a pattern matching phase, for fusing relevant attributes and providing a semantic interpretation of sensor data [Friedlander and Phoha 2002; Friedlander 2005; Whitehouse et al. 2006].

To the best of our knowledge, Friedlander and Phoha [2002] are the ones who introduced the concept of Semantic Information Fusion, which was applied for target classification. This work is further extended by Friedlander [2005] who describes techniques for extracting semantic information from sensor networks. The idea is to integrate and convert sensor data into formal languages. Then, the resulting language, obtained from the environment observations, is compared with the languages with known behaviors stored in a knowledge base. The idea behind this strategy is that behaviors represented by similar formal languages are semantically similar. Thus, this method extends traditional pattern-matching techniques that measure the distances between the feature vectors of an observed entity and a set of known behaviors.

Friedlander [2005] applies the proposed techniques to recognize the behavior of robots based on their trajectories, but they can also be used for saving resources. For instance, energy can be saved by making sensor nodes transmit only the formal language describing the perceived data, rather than every raw sensor data. Then at the external processing entity, the formal language can be used to classify the application behavior or to generate sensor data that are statistically equivalent to the original observations.

In any case it is necessary to have a set of known behaviors stored in a database, which in some cases may be difficult to obtain.

In another approach, Whitehouse et al. [2006] describe the Semantic Streams framework, which allows the user to formulate queries over semantic values without addressing which data or operations are to be used. Thus, the query answers are semantic interpretations acquired by in-network inference processes. Parallel to that work, Liu and Zhao [2005] propose the SONGS architecture, which by means of automatic service planning, converts declarative queries into a service composition graph, and performs optimizations for resource-aware execution of the service composite. These optimizations may include the avoidance of redundant computation of shared tasks that compose the queries issued by the user [Liu et al. 2005].

4.2. Estimation

Estimation methods were inherited from control theory and use the laws of probability to compute a process state vector from a measurement vector or a sequence of measurement vectors [Bracio et al. 1997]. In this section, we present the estimation methods known as: Maximum Likelihood, Maximum A Posteriori, Least Squares, Moving Average filter, Kalman filter, and Particle filter.

4.2.1. Maximum Likelihood (ML). Estimation methods based on Likelihood are suitable when the state being estimated is not the outcome of a random variable [Brown et al. 1992].

In the context of information fusion, given x, the state being estimated, and $\mathbf{z} = (z(1), \ldots, z(k))$, a sequence of k observations of x, the likelihood function $\lambda(x)$ is defined as the *probability density function* (pdf) of the observation sequence $\mathbf{z}$ given the true value of the state x:

$$\lambda(x) = p\,(\mathbf{z} \mid x)\,. \tag{10}$$

Note that the likelihood function is no longer a pdf.

The Maximum Likelihood estimator (MLE) searches for the value of x that maximizes the likelihood function

$$\hat{x}(k) = \arg\max_{x} p\,(\mathbf{z} \mid x) \tag{11}$$

that can be obtained from empirical or analytical sensor models.

Xiao et al. [2005] propose a distributed and localized MLE that is robust to the unreliable communication links of WSNs. In this method, every node computes a local unbiased estimate that converges towards the global Maximum Likelihood solution. The authors further extended this method to support asynchronous and timely delivered measurements: measurements taken at different time steps that happen asynchronously in the network [Xiao et al. 2006b]. Other distributed implementations of MLEs for WSNs include the Decentralized Expectation Maximization (EM) algorithm [Nowak 2003] and the Local Maximum Likelihood Estimator [Blatt and Hero 2004] that relax the requirement of sharing all the data.

In the network tomography domain, Hartl and Li [2004] use the MLE to estimate per-node loss rates during the aggregation and reporting of data from source to sink nodes. Such a strategy may be useful, for example, for routing algorithms to bypass lossy areas.

The MLE is commonly used to solve location discovery problems. In this context, the method is often used to obtain accurate distance (or direction, angle) estimations that are used to compute the location of nodes [Patwari et al. 2003; Fang et al. 2005] or sources (targets) [Sheng and Hu 2005; Niu and Varshney 2006; Li et al.

2006; Chen et al. 2006a]. An example of "node location discovery" is the Knowledge-Based Positioning System (KPS) [Fang et al. 2005], which assumes a prior knowledge about the pdf of the nodes' deployment so that sensor nodes can use the MLE to estimate their locations by observing the group memberships of their neighbors. An example of "source location discovery" is the bird monitoring application described in Chen et al. [2006a], which uses an approximate MLE to process acoustic measurements and estimate the source direction-of-arrival and perform beamforming for signal enhancement. Then, the direction information is used to localize the birds while enhanced signals are used to classify the birds.

4.2.2. Maximum A Posteriori (MAP). This method is based on Bayesian theory; therefore it is used when the parameter x to be discovered is the outcome of a random variable with known pdf $p(x)$. The measurement sequence is characterized by the sensor model (conditional pdf of the measurement sequence).

In the context of information fusion, given x, the state being estimated, and $\mathbf{z} = (z(1), \ldots, z(n))$, a sequence of k observations of x, the Maximum A Posteriori estimator searches for the value of x that maximizes the posterior distribution function

$$\hat{x}(k) = \arg\max_{x} p\,(x \mid \mathbf{z})\,. \tag{12}$$

Both methods, Maximum Likelihood and Maximum A Posteriori, try to find the most likely value for the state x. However, the first method assumes that x is a fixed though unknown point of the parameter space, while the last takes x as the outcome of a random variable with prior pdf known. These two methods are equivalent when the prior pdf of x is not informative, for example, when $p(x)$ is Gaussian with $\sigma \to \infty$ [Brown et al. 1992].

Schmitt et al. [2002] use the MAP estimator to find the joint positions of mobile robots in a known environment, and track the positions of autonomously moving objects. The collision resolution algorithm proposed by Yuan and Kam [2004] to manage traffic between local detectors (e.g., source nodes) and a fusion center (e.g., a cluster-head) use a MAP estimator to compute the number of nodes that wish to transmit so these nodes properly update their retransmission probability.

Traditional approaches for a MAP estimator may be too costly to be employed in WSNs [Rachlin et al. 2006]. However, a couple of efficient distributed solutions for WSNs have been proposed. Shah et al. [2005] present a distributed implementation in which MAP estimators are found as the maximum of concave functions so that simple numerical maximization algorithms can be used. Saligrama et al. [2006] use a variant of belief propagation [Pearl 1988; Ihler et al. 2005] as collaboration strategy for distributed classification that reaches a consensus to the centralized MAP estimate.

4.2.3. Least Squares. This class comprises estimation methods based on Least Squares. In a nutshell, the Least Squares method is a mathematical optimization technique that searches for a function that best fits a set of input measurements. This is achieved by minimizing the sum of the square error between points generated by the function and the input measurements. Different square-error metrics can be used (minimized) such as the ordinary squared error [Brown et al. 1992], the Huber loss function [Rabbat and Nowak 2004], and the root mean squared error [Guestrin et al. 2004]. For didactic reasons, we briefly discuss the ordinary least squares method [Brown et al. 1992] in the following.

The Least Squares method is suitable when the parameter to be estimated is considered fixed. In contrast to the Maximum A Posteriori Probability, this method does not assume any prior probability. Here, the measurements are handled as a deterministic function of the state, like

$$z(i) = h(i, x) + w(i), \tag{13}$$

where h represents the sensor model and w a noise sequence, for a sequence of $1 \leq i \leq k$ observations. The Least Squares method searches for the value of x that minimizes the sum of the squared errors between actual and predicted observations:

$$\hat{x}(k) = \arg\min_{x} \sum_{i=1}^{k} [z(i) - h(i, x)]^2. \tag{14}$$

Least Squares and Maximum Likelihood methods are equivalent when the noise $w(i)$ is a sequence of outcomes of independent identically distributed random variables with a symmetric zero-mean pdf [Brown et al. 1992].

Regarding WSNs, distributed implementations of the ordinary Least Squares and the Huber loss function are contrasted by Rabbat and Nowak [2004] who show that, in noisy environments, although the ordinary Least Squares algorithm quickly converges to the expected value, the variance is strongly affected by noisy measurements. This suggests that the Huber loss function is more suitable in many real cases in which noisy measurements might be frequent. To reduce communication, instead of transmitting the actual sensor data, Guestrin et al. [2004] share the parameters of a linear regression that describes the sensor data, and the values of these parameters are estimated by applying the Least Squares method with a root mean squared error as the optimization metric. Xiao et al. [2005; 2006b] use a weighted version of the Least Squares method to find an approximate solution for a distributed Maximum Likelihood estimation.

In another example, Willett et al. [2004] propose a spatial sampling algorithm in which a Least Squares method is used to define a small subset of sensor nodes that provide an initial estimate of the environment being sensed. This technique aims at building spatial maps describing properties of the sensor field [Nowak et al. 2004], and guiding mobile nodes in the construction of such maps [Singh et al. 2006].

Instead of transmitting the complete data stream from source to sink, Santini and Römer [2006] use a dual prediction scheme, based on Least Squares filters, both in the source and in the sink. Only when the predicted value differs from the actual value by more than a given error, the value is transmitted to the sink. Liu et al. [2006] propose a robust and interactive Least Squares method for node localization in which, at each iteration, nodes are localized by using a Least-Squares-based algorithm that explicitly considers noisy measurements.

4.2.4. Moving Average Filter. The moving average filter [Smith 1999] is widely adopted in digital signal processing (DSP) solutions because it is simple to understand and use. Furthermore, this filter is optimal for reducing random white noise while retaining a sharp step response. This is the reason that makes the moving average the main filter for processing encoded signals in the time domain. As the name suggests, this filter computes the arithmetic mean of a number of input measurements to produce each point of the output signal.

Given an input digital signal $\mathbf{z} = (z(1), z(2), \ldots)$, the true signal $\mathbf{x} = (\hat{x}(1), \hat{x}(2), \ldots)$ is estimated by

$$\hat{x}(k) = \frac{1}{M} \sum_{i=0}^{M-1} z(k - i), \tag{15}$$

for every $k \geq M$, where M is the filter's window, the number of input observations to be fused. Observe that M is also the number of steps the filter takes to detect the

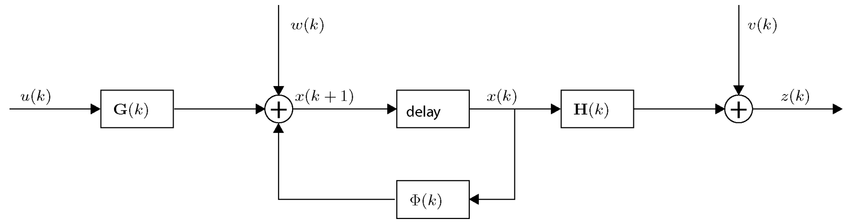

Fig. 3. Block diagram of the Kalman filter.

change in the signal level. The lower the value of M, the sharper the step edge. On the other hand, the greater the value of M, the cleaner the signal. When a step signal has random white noise, the Moving Average filter manages to reduce the noise variance by a factor $\sqrt{M}$ [Smith 1999]. Thus, M should be the smallest value in which this noise reduction meets the application requirements.

Woo et al. [2003] study the use of moving average filters within adaptive link estimators so that link connectivity statistics are dynamically collected and exploited by routing protocols to improve reliability. Nakamura et al. [2005b] use the Moving Average filter to estimate the data traffic of continuous WSNs; and that estimate is further used for routing-failure detection. Yang et al. [2005a] apply the Moving Average filter on target locations to reduce errors of tracking applications in WSNs. In the NED algorithm [Jin and Nittel 2006], sensor nodes estimate events and event boundaries based on simple Moving Average filters that are used to improve the sensor readings.

Weighted Moving Average filters are also commonly used in WSNs, especially the Exponentially Weighted Moving Average (EWMA) filter. An EWMA filter has multiplying factors to give different weights, which decrease exponentially, to different data points. EWMA filters have been used by MAC protocols to estimate ambient noise [Polastre et al. 2004] and determine if the channel is clear; and for local clock synchronization [Rhee et al. 2005] used for contention purposes. Applications have used EWMA filters to obtain refined estimations from sensors for detection and classification tasks [Gu et al. 2005] and to estimate distances for localization algorithms [Blumenthal et al. 2006]. Another example is the use of EWMA filters to detect incipient congestion [Rangwala et al. 2006] for fair and efficient rate control. Due to the increasing popularity of Moving Average filters, Tinker [Elson and Parker 2006], a high-level tool for application development in WSNs, includes a time-efficient implementation of the EWMA filter.

4.2.5. Kalman Filter. The Kalman filter is a very popular fusion method. It was originally proposed in 1960 by Kalman [1960] and it has been extensively studied since then [Luo and Kay 1992; Jacobs 1993; Brown and Hwang 1996].

The Kalman filter (depicted in Figure 3) estimates the state x of a discrete-time controlled process that is ruled by the state-space model

$$x(k+1) = \Phi(k)x(k) + \mathbf{G}(k)u(k) + w(k), \tag{16}$$

with measurements z represented by

$$z(k) = \mathbf{H}(k)x(k) + v(k), \tag{17}$$

where $\Phi(k)$ is the state transition matrix, $\mathbf{G}(k)$ is the input transition matrix, $u(k)$ is the input vector (e.g., the location of sensor platform), $\mathbf{H}(k)$ is the measurement matrix; w

and v are random variables obeying zero-mean Gaussian laws with covariance matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$, respectively.

Based on the measurement $z(k)$ and on the knowledge of the system parameters, the estimate of $x(k)$, represented by $\hat{x}(k)$, and the prediction of $x(k+1)$, represented by $\hat{x}(k+1 \mid k)$ are

$$\hat{x}(k) = \hat{x}(k \mid k-1) + \mathbf{K}(k)[z(k) - \mathbf{H}(k)\hat{x}(k \mid k-1)], \tag{18}$$

$$\hat{x}(k+1 \mid k) = \Phi(k)\hat{x}(t \mid t) + \mathbf{G}(k)u(k), \tag{19}$$

respectively, where $\mathbf{K}$ is the Kalman filter gain determined by

$$\mathbf{K}(k) = \Pr(k \mid k-1)\mathbf{H}^T(k)[\mathbf{H}(k)P(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k)]^{-1}, \tag{20}$$

where $P(k \mid k-1)$ is the prediction covariance matrix that can be determined by

$$P(k+1 \mid k) = \Phi(k)\Pr(k)\Phi^T(k) + \mathbf{Q}(k), \tag{21}$$

with

$$P(k) = P(k \mid k-1) - \mathbf{K}(k)\mathbf{H}(k)P(k \mid k-1). \tag{22}$$

The Kalman filter is used to fuse low-level redundant data. If a linear model can describe the system and the error can be modelled as Gaussian noise, the Kalman filter recursively retrieves statistically optimal estimates [Luo and Kay 1992]. However, to deal with nonlinear dynamics and nonlinear measurement models, other methods should be adopted. According to Jazwinski [1970], the variation named Extended Kalman filter (EKF) [Welch and Bishop 2001] is a popular approach to implement recursive nonlinear filters. More recently, the Unscented Kalman Filter (UKF) [Julier and Uhlmann 1997] has gained attention since it does not have a linearization step and the associated errors. The UKF uses a deterministic sampling technique to choose a minimal set of sample points around the mean. These points are propagated through the nonlinear functions so the covariance of the estimate is recovered. The standard Kalman Filter can be further extended to improve its performance [Gao and Harris 2002] or to provide decentralized implementations [Grime and Durrant-Whyte 1994].

In WSNs, we can find schemes to approximate distributed Kalman filtering, in which the solution is computed based on reaching an average consensus among sensor nodes [Spanos et al. 2005; Olfati-Saber 2005]. An important concern is data loss due to the unreliable communication channels in WSNs. In this context, Sinopoli et al. [2004] assess the performance of the Kalman filter in a scenario with intermittent observations and show the existence of a critical value for the arrival rate of the observations, beyond which the Kalman filter becomes unstable.

Another issue regarding the use of a Kalman filter in WSNs is that it requires a proximate clock synchronization among sensor nodes [Ganeriwal et al. 2003]. This is evidenced by Manzo et al. [2005] who show how synchronization problems caused by an attack on the time synchronization can affect the Kalman filter performance, leading to incorrect estimates.

For a long time, Kalman filters have been used in algorithms for source localization and tracking, especially in robotics [Brown et al. 1992]. Wireless sensor networks inherited this application trend and, aiming at accuracy improvement, the Kalman filter has been applied to refine location and distance estimates [Savvides et al. 2003; Hongyang et al. 2005], and track different sources [Li et al. 2006]. In particular, Li et al. [2006]

propose a source localization algorithm for a system equipped with asynchronous sensors, and show that the UKF outperforms, the EKF for source tracking because of the linearization error present in the EKF.

A MAC protocol can also benefit from the applicability of a Kalman filter to predict, for instance, its frame size. In this direction, Ci et al. [2004] use the UKF for frame size prediction, while Raviraj et al. [2005] use the EKF for the same purpose. As a conclusion, Ci and Sharif [2005] show that the UKF approach is better than the EKF, especially under noisy conditions.

Still in the context of data communication, Jain et al. [2004] use a dual Kalman Filter approach in which both source and sink nodes predict the sensed value so the source node sends data only when it knows the sink prediction is incorrect. In the SCAR routing algorithm [Mascolo and Musolesi 2006], a sensor node uses the Kalman filter to predict context information (mobility and resources) about its neighbors, and based on such predictions it chooses the best neighbor for routing its data.

4.2.6. Particle Filter. Particle filters are recursive implementations of statistical signal processing [Gilks et al. 1996] known as sequential Monte Carlo methods (SMC) [Crisan and Doucet 2002]. Although the Kalman filter is a classical approach for state estimation, particle filters represent an alternative for applications with non-Gaussian noise, especially when computational power is rather cheap and sampling rate is slow [Nordlund et al. 2002].

Particle filters attempt to build the posterior pdf based on a large number of random samples, called particles. The particles are propagated over time, sequentially combining sampling and resampling steps. At each time step, the resampling is used to discard some particles, increasing the relevance of regions with high posterior probability.

In such a filtering process, multiple particles (samples) of the same state variable x are used, and each particle has an associated weight that indicates the particle quality. Then, the estimate is the result of the weighted sum of all particles. The Particle filter algorithm has two phases: prediction and update. In the prediction phase, each particle is modified according to the existing model, including the addition of random noise in order to simulate the effect of noise. Then, in the update phase, the weight of each particle is reevaluated based on the latest sensory information available, so that particles with small weights are eliminated (resampling process).

Arulampalam et al. [2002] discuss the use of particle filters and the extended Kalman filter for tracking applications. Further analysis comparing the use of the extended Kalman filter and particle filters for state estimation is provided by Yuen and MacDonald [2002]. Zeng and Ma [2002] propose active particle filtering where every particle is first driven to its local maximum of the likelihood before it is weighted; as a result, the efficiency of every particle is improved and the number of required particles is reduced. Other examples of particle filters in the information fusion domain include applications for computer vision [Isard and Blake 1996], multitarget tracking [Doucet et al. 2002], and location discovery in wireless networks [Gunnarsson and Gustafsson 2003].

In WSNs, target tracking is currently the principal research problem wherein particle filters have been used. Aslam et al. [2003] propose a tracking algorithm based on particle filtering, which explores geometric properties of a network composed of sensors using a binary detection model (one bit representing whether a target is moving toward or away from the sensor). Coates [2004] investigates the use of distributed particle filters for target tracking within hierarchical networks in which the cluster-heads are responsible for computation and information sharing, while remaining cluster members are responsible for sensing only. Wong et al. [2004] also adopt a hierarchical collaborative data fusion scheme based on particle filters for cross-sensor (information from multiple sensors) fusion and cross-modality (information from different sensing modes) fusion for target tracking. Guo and Wang [2004] propose a novel SMC solution for target tracking that makes use of an auxiliary particle filter technique for data fusion, and a reduced representation of the a posteriori distribution, to reduce the amount of data transmitted among sensor nodes.

In contrast to single target tracking, multiple target tracking is a more difficult and more general problem, whose solutions may also use particle filters. Sheng et al. [2005] propose two distributed particle filters for multiple target tracking that run on uncorrelated sensor cliques that are dynamically organized based on target trajectories. Vercauteren et al. [2005] propose a collaborative solution based on the SMC methodology for jointly tracking several targets and classifying them according to their motion pattern. By using range data, Chakravarty and Jarvis [2005] propose a real-time system based on particle filters, for tracking an unknown number of targets, that incorporates a clustering algorithm to discern legitimate from fake targets. Kreucher et al. [2005] propose a particle filter algorithm that explicitly enforces the multiple target nature of the problem. The algorithm estimates the number and states of a group of moving targets occupying a surveillance region.

Another natural application of particle filters within WSNs is to find the nodes' locations. In this context, Hu and Evans [2004] use the particle filter for obtaining nodes' locations in a network composed of mobile nodes. The proposed solution works as a tracking solution applied to all nodes. Interestingly, the authors show that, despite the contrary intuition, mobility can improve the accuracy and reduce the costs of localization. Miguez and Artes-Rodriguez [2006] propose a Monte Carlo method for joint node location and target tracking that uses a particle filter for both target tracking and refinement of node position estimates.

Other interesting applications of particle filters include multiuser parameter tracking in communication systems [Guo et al. 2005] based on code division multiple access (CDMA), and blind symbol detection of orthogonal frequency-division multiplexing (OFDM) systems [Yang et al. 2005b]—a digital modulation scheme for high-rate wireless communications.

4.3. Feature Maps

For some applications, such as guidance and resource management, it might not be feasible to directly use raw sensory data. In such cases, features representing aspects of the environment can be extracted and used by the application. Commonly, diverse fusion methods of estimation and inference can be used to generate a feature map. Here, we explore two special types of feature maps: occupancy grid and network scans.

4.3.1. Occupancy Grid. Occupancy grids, also called occupancy maps or certainty grids, define a multidimensional (2D or 3D) representation of the environment, describing which areas are occupied by an object and/or which areas are free spaces. According to Elfes [1989], an occupancy grid is "a multidimensional random field that maintains stochastic estimates of the occupancy state of the cells": the observed space is divided into square or cubic cells and each cell contains a value indicating its probability of being occupied. Usually, such probability is computed—based on information provided by several sensors—using different methods, such as Bayesian theory, Dempster-Shafer reasoning, and fuzzy set theory [Ribo and Pinz 2001].

Occupancy grids were initially used to build an internal model of static environments based on ultrasonic data [Elfes 1987], and since then several variations have been proposed. Arbuckle et al. [2002] introduce the *temporal occupancy grid* as a method to model and classify spatial areas according to their time properties. Hoover and Olsen [1999, 2000] use a 2D raster as an occupancy map where each map pixel contains a binary value indicating if the respective space is occupied or empty.

Typical applications of occupancy grids include position estimation [Wongngamnit and Angluin 2001], robot perception [Hoover and Olsen 2000] and navigation [Pagac et al. 1998]. There are also applications in computer graphics, such as simulation of graphical creatures behavior [Isla and Blumberg 2002] and collisions detection of volumetric objects [Gagvani and Silver 2000].

4.3.2. Network Scans. Network Scans are defined by Zhao et al. [2002b] as a sort of resource/activity map for wireless sensor networks. Analogous to a weather map, the network scan depicts the geographical distribution of resources or activity of a WSN. By considering a resource of interest, instead of providing detailed information about each sensor node in the network, these scans offer a summarized view of the resource distribution. The network scan implemented by Zhao et al. [2002b] is called eScan and it retrieves information about the residual energy in the network in a distributed *in-network* fashion.

The algorithm is quite simple. First, an aggregation tree is formed to determine how the nodes will communicate. Second, each sensor computes its local eScan and whenever the energy level drops significantly since the last report, the node sends its eScan towards the sink. The eScans are aggregated whenever a node receives two or more topologically adjacent eScans that have the same or similar energy level. The aggregated eScan is a polygon corresponding to a region, and the summarized residual energy of the nodes within that region. Each energy level is assigned a gray level and the result is a 2D image (map) where white regions have nodes with full charge and black regions have dead nodes.

Although this algorithm makes unlikely assumptions for sensor networks, such as a perfect MAC (Medium Access Control) layer with no loss or overhead due to contention or environment changes, the network scan poses an interesting fusion method to present information about the network resources and activity. In the particular case of eScan, it allows the identification of low energy regions, helping designers decide where new sensors should be deployed. In addition, the network may use eScans to reorganize itself, so nodes with low energy levels are spared.

4.4. Reliable Abstract Sensors

In this section, we present information fusion methods especially proposed to deal with reliable abstract sensors. The concept of reliable abstract sensor was introduced by Marzullo [1990] to define one of three types of sensors: concrete, abstract, and reliable abstract sensors. A concrete sensor is the device that perceives the environment by sampling a physical state variable of interest. The abstract sensor is an interval of values that represents the observation provided by a concrete sensor. Finally, the reliable abstract sensor is the interval (or a set of intervals) that always contains the real value of the physical state variable. A reliable abstract sensor is computed based on several abstract sensors. Fusion methods for reliable abstract sensors have been used in the context of time synchronization so that sensor nodes perform external synchronization by maintaining lower and upper bounds on the current time [Römer et al. 2005].

4.4.1. Fault-Tolerant Averaging. The fault-tolerant averaging algorithm was first introduced by Marzullo [1984] in the context of time synchronization in distributed systems. Afterwards, it was used in the information fusion domain [Marzullo 1990] to fuse a set of n abstract sensors into a reliable abstract sensor that is correct even when some of the original sensors are incorrect.

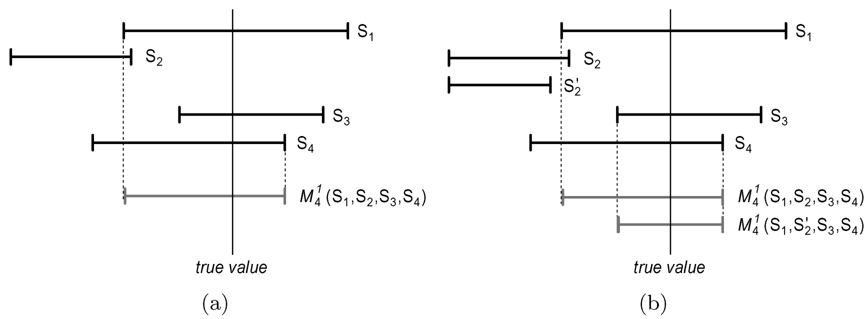

Fig. 4. Example of the Fault-Tolerant Averaging algorithm.

The algorithm assumes that, at most, f of n abstract sensors are faulty (i.e., incorrect) where f is a parameter. Let $\mathcal{I} = \{I_1, \ldots, I_n\}$ be the set of intervals $I_i = [x_i, y_i]$ provided by n abstract sensors referring to samples of the same physical state variable taken at the same instant. Considering that at most f out of n sensors are faulty, the fault-tolerant averaging computes $\mathcal{M}_n^f(\mathcal{I}) = [low, high]$, where low is the smallest value in at least $n - f$ intervals in $\mathcal{I}$, and $high$ is the largest value in at least $n - f$ intervals in $\mathcal{I}$. Marzullo [1990] shows that the algorithm has $O(n \log n)$ complexity.

As the algorithm computes an intersection of intervals, depending on the intervals in $\mathcal{I}$, the result $\mathcal{M}_n^f(\mathcal{I})$ can be more accurate than any sensor in $\mathcal{I}$, that is, the resultant interval can sometimes be tighter than the original ones. However, $\mathcal{M}_n^f(\mathcal{I})$ cannot be more accurate than the most accurate sensor in $\mathcal{I}$ when $n = 2f + 1$.

The result of $\mathcal{M}$ certainly contains the correct value when the number of faulty sensors is at most f. However, it may present an unstable behavior in the sense that minor changes in the input may produce quite different outputs.

Figure 4(a) depicts a scenario with four sensors $\{S_1, S_2, S_3, S_4\}$ with a faulty one. In this example, S_2 and S_3 do not have any intersection, consequently one of them is the faulty sensor. Since it is not possible to discover which one provides the correct interval, both must be covered to securely include the *true value*. Thus $\mathcal{M}_4^1(S_1, S_2, S_3, S_4)$ returns the interval [*low*, *high*], where low is the smallest value in at least $n - f = 4 - 1 = 3$ intervals (which is the left edge of S_1), and $high$ is the largest value in at least $n - f = 4 - 1 = 3$ intervals (which is the right edge of S_4).

Figure 4(b) illustrates the instability of $\mathcal{M}$. In this case, if the right edge of S_2 moves to the left, as given by S_2', then the left edge of the result becomes the left edge of S_3. Thus a small change in S_2, but large enough to avoid an intersection with S_1, causes a great variation in the final result.

Chew and Marzullo [1991] extend the original single-dimensional fault-tolerant averaging algorithm to fuse data from multidimensional sensors. Another extension of Marzullo's original work is provided by Jayasimha [1994], who improves the detection of faulty sensors for the linear case.

Blum et al. [2004] show the worst-case (when all clocks run with maximal drift) optimality of the $\mathcal{M}$ function, and propose an improved algorithm, the Back-Path Interval Synchronization Algorithm (BP-ISA), which also is worst-case-optimal but yields better results in the average case wherein every node stores, maintains, communicates, and uses the bounds from its last communication with other nodes. In this context, Meier et al. [2004] show that, although optimal interval-based synchronization can only be achieved by having nodes store and communicate their entire history, efficient average-case-optimal synchronization can be obtained by using only recent data.

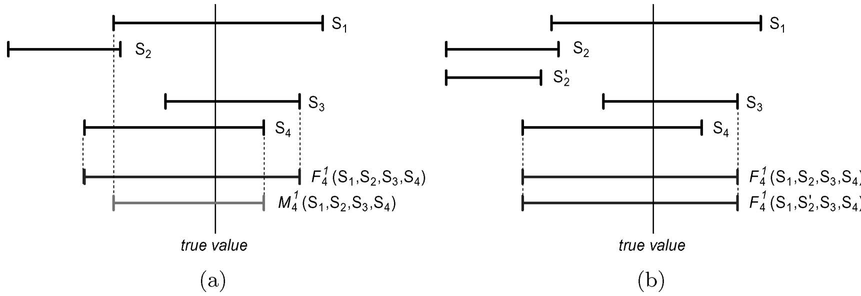

Fig. 5. Example of the Fault-Tolerant Interval function.

4.4.2. The Fault-Tolerant Interval Function. The Fault-Tolerant Interval (FTI) function, or simply the $\mathcal{F}$ function, was proposed by Schmid and Schossmaier [2001]. FTI is an alternative integration function that considers the width of the intervals being fused.

The algorithm also assumes that, at most, f of n abstract sensors are faulty where f is a parameter. Let $\mathcal{I}$ be the set of intervals provided by n abstract sensors, as defined in Section 4.4.1. The FTI intersection function is $\mathcal{F}_n^f(\mathcal{I}) = [low, high]$, where low corresponds to the $(f+1)^{th}$ largest of the left edges $\{x_1, \ldots, x_n\}$, and $high$ is the $(f+1)^{th}$ smallest of the right edges $\{y_1, \ldots, y_n\}$.

The $\mathcal{F}$ function is robust. This means that it assures that minor changes in the input intervals will result in minor changes in the integrated result. Schmid and Schossmaier [2001] show that minor changes in the input intervals might greatly change the result of the Marzullo's Fault-Tolerant Averaging.

To illustrate the behavior of $\mathcal{F}$ we consider in Figure 5, the same example provided before (in Figure 4). The resulting interval is slightly larger than the intervals returned by $\mathcal{M}$ (Figure 5(a)). However, the resulting interval does not change when S_2' is used instead of S_2 (Figure 5(b)). As a general result, $\mathcal{M}$ tends to achieve tighter intervals than $\mathcal{F}$. However, $\mathcal{F}$ is less vulnerable to small changes in the input intervals.

Although the experiments analysis of Blum et al. [2004] and Meier et al. [2004] (see Section 4.4.1) consider the $\mathcal{M}$ function as a proof-of-concept, the authors' findings are also extensible to the $\mathcal{F}$ function.

4.5. Aggregation

Kulik et al. [2002] define data aggregation as a technique used to overcome two problems: *implosion* and *overlap*. In the former, data sensed by one node is duplicated in the network due to the data routing strategy (e.g., flooding). The *overlap* problem happens when two different nodes disseminate the same data. This might occur when the sensors are redundant—they sense the same property in the same place. In both cases, redundancy, which occurs due to different reasons, might have its negative impact (e.g., waste of energy and bandwidth) reduced by data aggregation and information fusion.

Aggregation techniques are the common summarization functions used by query languages (e.g., SQL) to retrieve summarized data in database systems [Madden et al. 2002]. The use of data aggregation in WSNs and its impact on energy consumption is the subject for further research. Krishnamachari et al. [2002] provide theoretical results regarding the NP-completeness related to the formation of an optimal aggregation tree. Intanagonwiwat et al. [2002] evaluate the impact (latency and robustness) of a greedy aggregation algorithm in high-density networks. Boul et al. [2003a] discuss the trade-off between energy consumption and accuracy when aggregation functions are used to summarize data from a WSN. The seminal TinyDB [Madden et al. 2005] is a distributed query processor that offers simple extensions to SQL for controlling data acquisition and allowing the user to specify temporal and event-based aggregates.

Other aggregation functions can be identified in WSNs, which are *suppression* [Intanagonwiwat et al. 2000] and *packaging* [He et al. 2004]. The former function simply suppresses redundant data by discarding duplicates. For example, if a node senses the temperature 45°C and receives the same observation from a neighbor, then only one packet containing a 45°C observation will be forwarded. The second aggregation function groups several observations in one single packet. The objective of this strategy is to avoid the overhead of the MAC protocol when sending several packets. However, *packaging* may not be classified as a fusion technique because it does not exploit the synergy among the data. *Packaging* is actually a solution to optimize the usage of a communication protocol, which is independent of any fusion method.

4.6. Compression

Classical compression techniques, such as the Ziv-Lempel and Huffman families [Nelson and Gailly 1995], are not information fusion methods, as they consider only the coding strategy used to represent data regardless of their semantics. However, in WSNs, data can be compressed by exploiting spatial correlation among sensor nodes in a distributed fashion demanding no extra communication except the dissemination of the sensed data [Hoang and Motani 2005a; 2005b]. This is possible by considering that two neighbors provide correlated measurements (observations). In this section, we include the compression methods that exploit the synergy among the sources to achieve smaller codes that would not be possible if any of these sources were used individually.

4.6.1. Distributed Source Coding. Distributed Source Coding (DSC) [Xiong et al. 2004] refers to the compression of multiple correlated sources, physically separated, that do not communicate with each other (thus distributed coding). These sources can send their compressed outputs to a central unit (e.g., a sink node) for joint decoding. Kusuma et al. [2001] and Pradhan et al. [2002] pioneered the use of DSC for data compression in WSNs by proposing the Distributed Source Coding Using Syndromes (DISCUS) framework. Thus, we will use DISCUS to illustrate DSC in WSNs.

Distributed Source Coding Using Syndromes (DISCUS) is a constructive framework that addresses the problem of distributed data compression for WSNs. The main idea is that when a sensor node A needs to send its observation to a correlated sensor node B, it is not necessary to transmit all bits used to code A's observation.

To understand DISCUS, see the example illustrated in Figure 6, where a sensor observation is coded with 3-bit words. In this case, a sensor observation is a value in the set {000, 001, 010, 011, 100, 101, 110, 111}. Suppose that A and B are equiprobable 3-bit words correlated, such that the Hamming distance[2] between A and B is at most one, that is, the difference of A and B can be only one bit. Now the possible values for an observation are grouped into four cosets such that the Hamming distance among the elements of a coset is three: {000, 111}, {001, 110}, {010, 101}, {100, 011}. Node A can send only the index of the coset containing its observation, and B can decode this index based on the fact that the Hamming distance between its own observation and the one provided by A is at most one. Thus, if A senses 101 it can send to B only the index (10) of the coset {010, <u>101</u>}. When B receives the index 10 from A, it accesses

[2] The Hamming distance is the number of bits in two binary strings of equal length for which the corresponding elements are different.

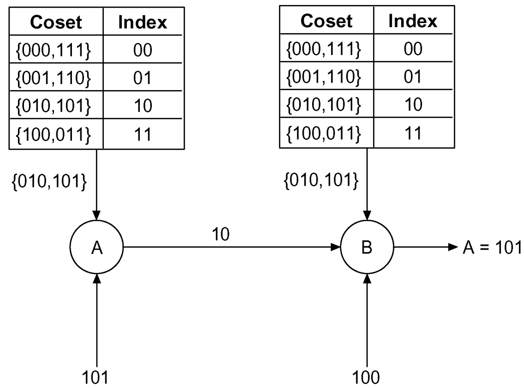

Fig. 6. Example of data compression for WSNs using DISCUS.

its own reading (100) and concludes that, as the Hamming distance from its own observation and the one provided by A is at most one, the value provided by A should be 101.

Details about the design and construction of DISCUS are presented by Pradhan and Ramchandran [2003]. Tang et al. [2003] propose a DSC scheme for data compression based on a cost function that considers the energy necessary for encoding, transmitting, and decoding the bitstream being compressed. Marco and Neuhoff [2004] study the impact of packet losses in compression schemes based on DSC, and try to characterize the tradeoff between the compression rate and the loss factor of such encoding schemes. Hua and Chen [2005] propose an improved Viterbi algorithm [Forney 1973] (decoder algorithm) for DSC based on convolutional and turbo codes, which takes advantage of known parity bits at the decoder for error correction. Zhang and Wicker [2005] provide a framework for the design and analysis of distributed, joint source and network coding [Ahlswede et al. 2000] algorithms that optimize the tradeoff between compression efficiency and network robustness. Motivated by the complexity of such a problem, Ramamoorthy et al. [2006] investigate whether source coding can be separated from network coding, and conclude that, in general, such a separation is not possible, but in the case of two sources and two receivers, a separable solution always exists. Complementarily, Barros and Servetto [2006] show that separation holds in a fairly general network situation that allows only independent point-to-point channels between pairs of nodes, and not multiple-access or broadcast channels.

4.6.2. Coding by Ordering. Petrovic et al. [2003] propose a compression strategy called Coding by Ordering. In this case, every node in a region of interest sends its data to one node in the region, called a border node, which is responsible for grouping all packets into a super-packet that will be sent towards the sink node. This strategy relies on the fact that when the packet order in the super-packet is unimportant, the border node can suppress some packets and sort the remainder, such that the values of the suppressed packets are indicated.

To illustrate how Coding by Ordering works, see the following example. There are four nodes A, B, C, and D. Each node provides an observation, which is a value between 0 and 5. Then, the border node can choose to suppress the value provided by node D, by choosing the proper ordering among the $3! = 6$ possible orderings of the packets from nodes A, B, and C according to the values in Table I. Thus, if node D observation is 0, then the super-packet ordering will be $\{A,B,C\}$; if its observation is 1, then the ordering will be $\{A,C,B\}$, and so on.

Table I. Example of Data Compression Using Coding by Ordering

Packet Ordering	Observation from node D
$\{A,B,C\}$	0
$\{A,C,B\}$	1
$\{B,A,C\}$	2
$\{B,C,A\}$	3
$\{C,A,B\}$	4
$\{C,B,A\}$	5

Although the Coding by Ordering scheme is simple, it does not explore the possible correlation among the sensor nodes as DISCUS does. Furthermore, in a number of practical cases, Coding by Ordering might be infeasible. For instance, considering the previous example, if the observation provided by the sensor nodes varied from 0 to 6, than it would not be possible to suppress the observation from node D, because the number of the possible orderings $3! = 6$ would be smaller than the number of possible values of an observation.

4.6.3. Other Data Compression Approaches. Different approaches and techniques can be used for compressing sensor data. The Easinet Packet Compression (EasiPC) [Ju and Cui 2005] is a packet compression mechanism for WSNs that explores the redundancy within a single packet to be transmitted. Further investigation about the possibility of combining packet compression and a distributed data compression (e.g., DSC) could lead to broader solutions. Compressive Wireless Sensing (CWS) [Bajwa et al. 2006] is a distributed source-channel communication architecture based on compressive sampling [Haupt and Nowak 2006], for energy efficient estimation of sensor data that contain structural regularity. Requiring no sensor collaboration, the Distributed Compressed Sensing (DCS) [Duarte et al. 2006] framework exploits both intra and inter signal correlation structures providing resiliency, scalability, adaptability, and reduced computational complexity at the sensors.

Wavelets have also been used for data compression in WSNs [Ciancio and Ortega 2004; Tang and Raghavendra 2005; Ciancio et al. 2006; Wagner et al. 2006]. Ciancio and Ortega [2004] propose a distributed wavelet algorithm, based on a lifting scheme, as a means to decorrelate data at the nodes by exchanging information among neighbors. This work is extended by Ciancio et al. [2006], who consider networks where each sensor can use different compression schemes, which include wavelet transforms or simpler approaches (quantization), and provide a framework that allows selecting the best scheme for each sensor. Tang and Raghavendra [2005] propose a source broadcast scheme that uses wavelets to extract significance bits from sensor data; and those bits are used for exploiting the spatial and temporal correlation present in sensor data. Wagner et al. [2006] propose a simple interpolatory wavelet transform for irregular sampling grids that is used for distributed lossy compression of sensor data.

The integration of compression and routing techniques should be carefully studied as stated by Scaglione and Servetto [2002], whose approach is not based on distributed coding techniques, but on the use of classical source codes combined with suitable routing algorithms. Luo and Pottie [2005] study the integrated use of trees and clusters for data routing and Differential Pulse-Code Modulation (DPCM) for data compression. Similarly, Hoang and Motani [2005a; 2005b] exploit the inherent broadcast nature of wireless channels to jointly perform data compression and data routing in a cluster-based network.

There has been recent focus on the problem of joint data compression and parameter estimation [Pattem et al. 2004; Fowler and Chen 2005; Xiao et al. 2006a; Rabbat et al.

2006]. In particular, Rabbat et al. [2006] provide a broader solution, which is a distributed compression and storage system for sensor networks that allows a user, from any point in the network, to obtain sufficient information to reconstruct an accurate approximation of the entire network by querying a small number of nodes. An interesting result regarding data routing and data compression is presented by Pattem et al. [2004], who show that, although optimal routing with compression clearly depends on the data correlation level, there is a practical static clustering scheme that can provide near-optimal performance for a wide range of spatial correlations.

4.7. An Information Theory Approach

We know from the theory of distributed detection that we may have a higher reliability and a lower probability of detection error whenever data from multiple distributed sources are fused using a decision-making algorithm, rather than using a single observation source [Varshney 1997]. As expected, a network comprised of multiple sensors can increase data reliability and increase confidence in sensor observations. On the other hand, data generated by a single sensor are only approximations of the actual environment state and are restricted by the spatial and physical limitations of the sensing device.

Information processed in a network with multiple sensors represents observations from the environment in which those sensors are embedded, that are in general, difficult to estimate in advance. Thus, the process of collecting and processing data is typically probabilistic, which can be quantified using the principles of information theory [Varshney 1997]. Furthermore, the detection, processing and fusion of observations in a WSN are the basic elements of classical statistical decision theory [Poor 1994].

Both information theory and detection theory can be used to tackle the problem of transmitting and receiving information, as well as the more general problem of data fusion in distributed systems. Ahmed and Pottie [2005] propose using a Bayesian probabilistic approach for data fusion in WSNs. This framework allows weighing and processing observations obtained from different types of sensors (e.g., acoustic and magnetic) with different sensing capabilities (e.g., sensing mode and signal to noise ratio) in a systematic way. Tradeoffs in information rate and distortion can be determined using entropies, as addressed in rate distortion theory [Gallager 1968]. This is an interesting and promising approach that still needs to overcome some difficulties such as the uncertainty in measurements, the entropy of the source nodes, and how to efficiently convert sensor observations into entropies.

5. ARCHITECTURES AND MODELS

Several architectures and models have been proposed to serve as guidelines to design information fusion systems. This section presents the evolution of the models and architectures for such systems. Chronologically, these models evolved from information-based models to role-based models.

These models are useful for guiding the specification, proposal, and usage of information fusion within WSNs. As we show in the following, some of these models, such as the JDL and Frankel-Bedworth, provide a systemic view of information fusion, whereas others, such as the Intelligent Cycle and the Boyd Control Loop, provide a task view of information fusion.

5.1. Information-Based Models

Models and architectures proposed to design information fusion systems can be centered on the abstraction of the data generated during fusion. This section discusses the

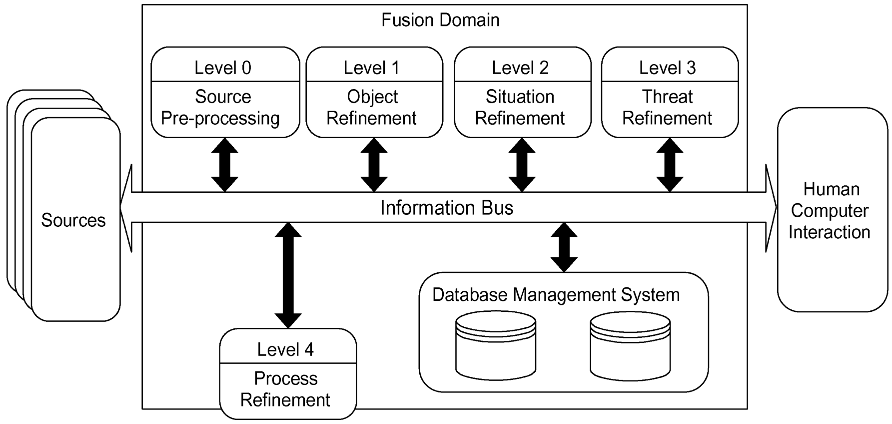

Fig. 7. The JDL model, figure from Hall and Llinas [1997].

models that specify their stages based on the abstraction levels of information manipulated by the fusion system.

5.1.1. JDL Model. JDL is a popular model in the fusion research community, commented on and revised in references such as Steinberg et al. [1999]. It was originally proposed by the U.S. Joint Directors of Laboratories (JDL) and the U.S. Department of Defense (DoD) [Kessler et al. 1992]. The model is composed of five processing levels, an associated database, and an information bus connecting all components. Its structure is depicted in Figure 7 and its components are described as follows:

Sources. Sources are responsible for providing the input information, and can be sensors, a priori knowledge (e.g., reference and geographical information), databases, or human input.

Database Management System. This system supports the maintenance of the data used and provided by the information fusion system. This is a critical function as it supposedly handles a large and varied amount of data. In WSNs, this function might be simplified to fit the sensors' restrictions of resources. Central to this issue is the proposal of data-centric storage systems that allow the network to efficiently answer queries without the need for directly querying all sensor nodes. Such a system stores data by name into a node (or a set of nodes) so that when the user (or another sensor node) needs data, it may directly query the node storing that type of data [Ratnasamy et al. 2003; Li et al. 2003; Gummadi et al. 2005; Sheng et al. 2006; Ahn and Krishnamachari 2006].

Human Computer Interaction (HCI). HCI is a mechanism that allows human input, such as commands and queries, and the notification of fusion results through alarms, displays, graphics, and sounds. Commonly, human interaction with WSNs occurs through query-based interfaces such as the ones used by Cougar [Yao and Gehrke 2002], TiNA [Sharaf et al. 2003], and TinyDB [Madden et al. 2005] projects.

Level 0 (Source Preprocessing). Also referred to as Process Alignment, this level aims to reduce the processing load by allocating data to appropriate processes and selecting appropriate sources. In WSNs, source selection is a key issue for achieving intelligent resource usage while keeping the quality of information fusion. In this context, Zhao et al. [2002a; 2003a] propose an information-directed approach in which sources are chosen by dynamically optimizing the information utility of data for a given cost of communication and computation.

Level 1 (Object Refinement). Object refinement transforms the data into a consistent structure. Source localization, and therefore, all tracking algorithms are in Level 1, since they transform different types of data, such as images, angles, and acoustic data, into target locations. Section 4.2 presents some examples of tracking algorithms.

Level 2 (Situation Refinement). Situation refinement tries to provide a contextual description of the relationship among objects and observed events. It uses a priori knowledge and environmental information to identify a situation. As an example, Chen et al. [2006a] observe the acoustic signals from birds and, based on a predefined set of bird sound patterns, a contextual description of the relationship between the collected acoustic signals and that pattern basis is provided so we can infer the bird's class.

Level 3 (Threat Refinement). Threat refinement evaluates the current situation projecting it into the future to identify possible threats, vulnerabilities, and opportunities for operations. This is a difficult task because it deals with computation complexities and "enemies" intent assessment. The prediction step of tracking algorithms is in Level 3. By identifying a target and predicting its future location, we can identify whether or not it represents a threat. Examples of tracking algorithms are discussed in Section 4.2.

Level 4 (Process Refinement). This is a meta-process[3] responsible for monitoring the system performance and allocating the sources according to the specified goals. This function may be outside the domain of specific data fusion functions. Therefore, it is shown partially outside the data fusion process. A WSN should be monitored continuously (e.g., by using the tools provided by Zhao et al. [2003b]), and should collect management information (e.g., energy maps [Mini et al. 2004]), and provide QoS (e.g., coverage [Meguerdichian et al. 2001b] and exposure information [Megerian et al. 2002]) to support source allocation. For instance, the SCAR routing algorithm [Mascolo and Musolesi 2006] uses resource information to choose the best node to forward a packet. Zhao et al. [2003a] may also use resource information to select sources in a target tracking application.

The JDL model was proposed for military research so its terminology and original application is defense-oriented. Another drawback of the JDL model is that it does not make explicit the interaction among the processing elements. Moreover, it suppresses any feedback: it does not specify how current or past results of fusion can be used to enhance future iterations.

The JDL model provides a systemic view of the network that performs information fusion. Therefore, it guides the designer through the identification of the major solutions to incorporate in the network. For instance, from this discussion, the project might include the TinyDB query system [Madden et al. 2005], a target tracking algorithm with information-driven source selection [Zhao et al. 2003a], the GHT data-centric storage system [Ratnasamy et al. 2003] for efficient data distribution, and an energy map [Mini et al. 2004] for resource management.

5.1.2. Dasarathy Model. The Dasarathy or DFD (Data-Feature-Decision) model [Dasarathy 1997] is a fine-grained information-centered model in which the elements of information fusion are specified based on their inputs and outputs. Figure 8 depicts the DFD model. The primary input is raw data and the main output is a decision. The components responsible for the several fusion stages are the elements DAI-DAO, DAI-FEO, FEI-FEO, FEI-DEO and DEI-DEO, described in Section 3.3.

The DFD model is successful in specifying the main types of fusion regarding their input and output data. For this reason it is also used to classify information fusion (see

[3] A meta-process is a process that deals with and manipulates other processes.

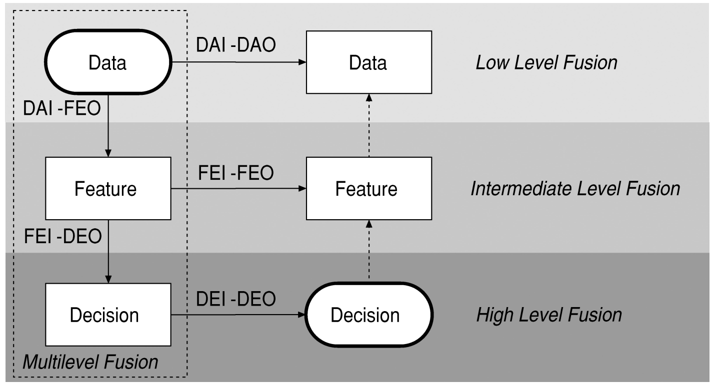

Fig. 8. The DFD model.

Section 3.3). Although it is not clear in Figure 8 how the architecture handles quality improvement, the system performance is enhanced by the decision blocks that use the system's feedback to tune its decision ability.

As Wald [1999] highlights, the input and output of a fusion process can be of any level (data, feature, decision). Therefore, the DFD model is limited in the sense that Dasarathy's functional blocks should be combined to provide more complex fusion blocks. For example, to provide a block that fuses a feature with two signals to obtain a refined feature, the signals should be fused by a DAI-FEO (Data In–Feature Out) block, then its output should be fused with the given feature by a FEI-FEO (Feature In–Feature Out) block.

In contrast to the JDL model, the DFD model does not provide a systemic view, instead it provides a fine-grained way to specify fusion tasks by means of the expected input and output data. Therefore, the DFD model is useful for specifying and designing fusion algorithms in WSNs with different purposes such as ambient noise estimation [Polastre et al. 2004] (DAI-DAO), feature map building [Singh et al. 2006] (FEI-FEO), event detection [Luo et al. 2006] (DEI-FEO), and failure detection [Nakamura et al. 2005b] (FEI-FEO).

5.1.3. Some Remarks on the Information-Based Models. Information-based models represent the first generation of models for information fusion, which focuses on the abstraction level of the information handled by the fusion tasks. In general, a limitation of such models is that they do not specify the execution sequence of the fusion tasks. Historically, the JDL model represents the first serious effort to provide a detailed model and a common terminology for the fusion domain. However, as it was born of military applications, the terminology adopted is threat-oriented.

The DFD model is possibly the most mature representative of these models. It is a fine-grained model that makes explicit the abstraction level of both input and output information of each fusion task. The DFD model differs from the JDL in the terminology adopted and in the approach used in the model. The JDL is oriented to military applications and the fusion tasks identified in the model reflect the peculiarities of such an application domain. On the other hand, the DFD model is oriented to the input and output of a fusion task regardless of the application domain. As a consequence, the specified functional blocks are "purely" focused on the fusion domain no matter the application. The key difference between the JDL and the DFD models is that the former provides a system-oriented perspective of information fusion—which is suitable for designing systems that incorporate fusion tasks—whereas the latter provides an

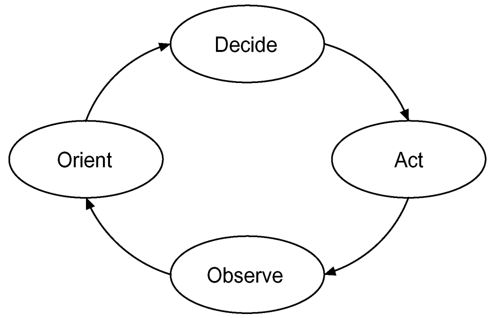

Fig. 9. The OODA loop.

input-output-oriented perspective of information fusion—which is suitable for understanding the relationship among fusion tasks and the manipulated data.

Within the WSN domain, these models can be used to facilitate understanding of the requirements and limitations introduced by fusion techniques. Although such models do not specify the network aspects (distributed nature) of WSNs, they work as a guide to specify which methods can be used and how they can be integrated with a given application.

5.2. Activity-Based Models

Some models are specified based on the activities that must be performed by an information fusion system. In such models, the activities and their correct sequence of execution are explicitly specified.

5.2.1. Boyd Control Loop. The Boyd Control Loop or OODA (*Observe, Orient, Decide, Act*) Loop [Boyd 1987] is a cyclic model composed of four stages (Figure 9). According to Bass [2000] this model is a representation of the classic decision-support mechanism of military information systems, and because such systems are strongly coupled with fusion systems, the OODA loop has been used to design information fusion systems. The stages of the OODA loop define the major activities related to the fusion process, which are:

Observe. Information gathering from the available sources.

Orient. Gathered information is fused to obtain an interpretation of the current situation.

Decide. Specify an action plan in response to the understanding of the situation.

Act. The plan is executed.

To exemplify the OODA model, let us consider the SCAR routing algorithm [Mascolo and Musolesi 2006]. In the SCAR algorithm, a sensor node collects initial neighborhood context information—mobility and resources—(*Observe* step) that feeds a Kalman filter used to predict future values and update current estimations (*Orient* step). Based on such predictions, the best neighbor is elected (*Decide* step), and the packet is forwarded to that node (*Act* step).

According to Bedworth and O'Brien [1999], *Observe* corresponds to level 0 (source preprocessing) of the JDL model; *Orient* encompasses levels 1, 2, and 3; *Decide* matches level 4 (process refinement); and *Act* is not dealt by the JDL model.

The OODA loop is a broad model that allows the specification and visualization of the system tasks in an ample way: it allows the modelling of the main tasks of a

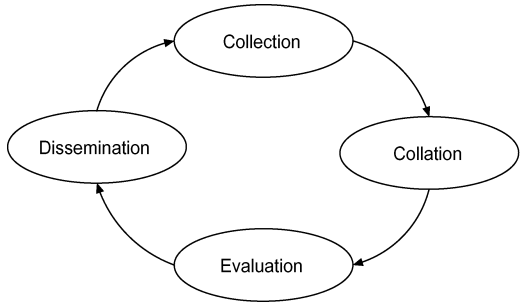

Fig. 10. The Intelligence Cycle.

system. However, OODA fails to provide a proper representation of specific tasks of an information fusion system.

5.2.2. Intelligence Cycle. The UK intelligence community describes the intelligence process as a four-stage cycle, which is called Intelligence Cycle [Shulsky and Schmitt 2002]. The Intelligence Cycle, depicted in Figure 10, describes the process of developing raw (sensory) information into finished intelligence used in decision-making and action. The stages (activities) of the Intelligence Cycle are:

Collection. Raw information is collected from the environment.

Collation. Collected information is analyzed, compared, and correlated. Irrelevant and unreliable information is discarded.

Evaluation. Collated information is fused and analyzed.

Dissemination. Fusion results are delivered to users who utilize the fused information to produce decisions and actions in response to the detected situation.

All tracking algorithms previously mentioned can be used to depict the Intelligence Cycle. For instance, Li et al. [2006] propose a source localization algorithm for a system equipped with asynchronous sensors wherein range data is collected (*Collection* step) and selected (*Collation* step). Then, the UKF is applied to estimate and predict the targets' locations (*Evaluation* step) that can be used to guide surveillance decisions (*Dissemination* step).

In comparison to the JDL model, *Collection* matches level 0 of the JDL model; *Collation* includes level 1; *Evaluation* comprises levels 2 and 3; and *Dissemination* corresponds to level 4 of the JDL model. Unlike the OODA model, the Intelligence Cycle does not make explicit the planning (*Decide*) and the execution (*Act*) phases, which are presumably included in the *Evaluation* and *Dissemination* phases. Again, this model does not to represent the specific tasks of an information fusion system.

5.2.3. Omnibus Model. Unlike the JDL model, the Omnibus model [Bedworth and O'Brien 1999] organizes the stages of an information fusion system in a cyclic sequence, just as the OODA loop and the Intelligence Cycle do. However, it makes explicit the activities referring to information fusion tasks (see Figure 11). The Omnibus model should be applied repetitively during the design phase of an information fusion system. Initially, it should be used to model the framework providing an overall perception of the system. Next, the model can be used to design the subtasks providing a fine-grained understanding of the system.

The Omnibus model was originally proposed to deal with information gathered by sensor devices. Some modifications can be suggested to make it more general and suitable for other information fusion systems. First, *Sensing* and *Signal Processing* can be

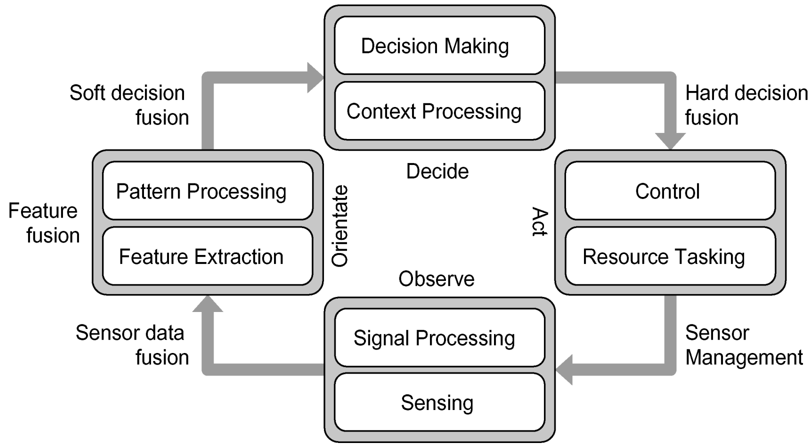

Fig. 11. Omnibus model [Bedworth and O'Brien 1999].

replaced by information gathering and information preprocessing, respectively. Second, *Sensor data fusion* should be stated as raw data fusion. Third, instead of *Sensor management* we should adopt source management. In this way, we make the Omnibus model suitable for information systems that deal with any kind of sources, including sensors.

The Omnibus model is essentially an enhanced version of the OODA loop specifically for the information fusion domain. To exemplify its usage in WSNs, let us consider a general surveillance application performing a target tracking task based on acoustic sensors. In the *Observe* stage, we could use, for instance, a moving average filter to reduce noise (*Signal Processing*) from acoustic sensor data (*Sensing*) provided by every sensor. In the *Orientate* stage, we translate the acoustic data into range estimation (*Feature Extraction*) and estimate the target's location and trajectory (*Pattern Processing*). In the *Decide* stage, we classify the sensed target (*Context Processing*) and evaluate whether that target represents a threat (*Decision Making*). If it is, in the *Act* state, we may respond by trying to intercept the target with a missile (*Control*), so we activate our arsenal (*Resource Tasking*). The tracking solutions previously discussed—such as Zhao et al. [2002a], Aslam et al. [2003], Chakravarty and Jarvis [2005], Sheng et al. [2005], Vercauteren et al. [2005], Sheng et al. [2005], and Miguez and Artes-Rodriguez [2006]—fit in this example.

5.2.4. Some Remarks on Activity-Based Models. The first two activity-based models (OODA and Intelligence Cycle) are general and can be employed in any application domain. As a result, they do not fulfill the specific aspects of the fusion domain demanding, thus, experience and expertise to model fine-grained fusion tasks. The last and most evolved representative, the Omnibus model, is not really a new model. In fact, it is the OODA model with a fine-grained structure regarding the fusion domain. It details the OODA activities specifying the fusion tasks that should be executed at each stage. The activities identified are similar to the ones provided by the Waterfall model [Markin et al. 1997] in such a way that the Omnibus model seems to be an integration of both, the Waterfall and the OODA models.

5.3. Role-Based Models

Role-based models represent a change of focus on how information fusion systems can be modelled and designed. In such models, information fusion systems are specified

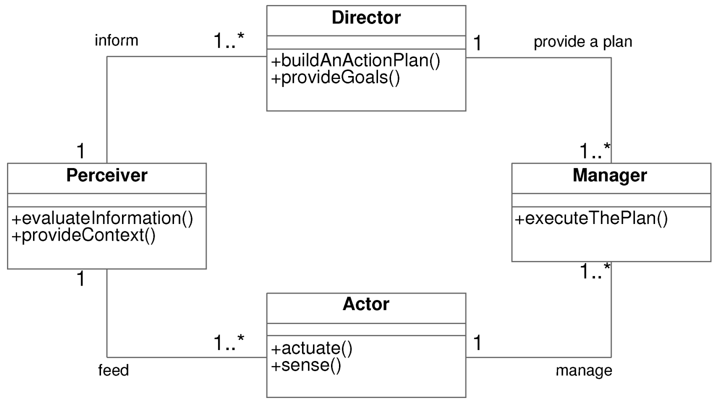

Fig. 12. The Object-Oriented model for information fusion, figure adapted from Kokar et al. [2000].

based on the fusion roles and the relationships among them providing a more fine-grained model for the fusion system. The two members of this generation are the Object-Oriented Model [Kokar et al. 2000] and the Frankel-Bedworth Architecture [Frankel and Bedworth 2000]. Like the JDL model, the role-based models herein also provide a systemic view of information fusion. However, in contrast to the previous models, role-based models do not specify fusion tasks or activities. Instead, they provide a set of roles and specify the relationships among them.

5.3.1. Object-Oriented Model. Kokar et al. [2000] propose an object-oriented model for information fusion systems. This model also uses cyclic architecture. However, unlike the previous models, it does not specify fusion tasks or activities. Instead, the object-oriented model provides a set of roles and specifies the relationship among them. Figure 12 is a simplification of the object-oriented model provided by Kokar et al. [2000] in which four roles are identified:

Actor. Responsible for the interaction with the world, collecting information and acting on the environment.

Perceiver. Once information is gathered, the perceiver assesses such information providing a contextualized analysis to the director.

Director. Based on the analysis provided by the perceiver, the director builds an action plan specifying the system's goals.

Manager. The manager controls the actors to execute the plans formulated by the director.

From the realization perspective (role of the objects), human and computer objects are not distinct. For this reason, the Object-Oriented model will not likely be directly mapped onto actual system implementations based on object-oriented programming languages. Nonetheless, it deserves this brief discussion, for it is an intermediate model towards the Frankel-Bedworth architecture [Frankel and Bedworth 2000] that we discuss below.

5.3.2. Frankel-Bedworth Architecture. Frankel [1999] describes an architecture for human fusion composed of two self-regulatory processes: local and global. Local estimation process manages the execution of the current activities based on goals and timetables provided by the global process. The global process updates the goals and timetables according to the feedback provided by the local process. Frankel's architecture is then

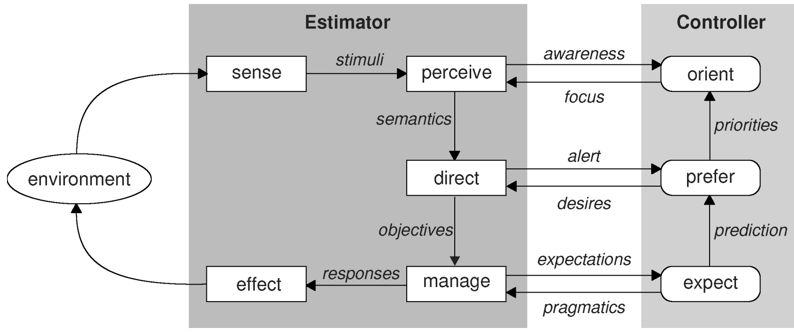

Fig. 13. The Frankel-Bedworth architecture, figure adapted from Frankel and Bedworth [2000].

transported to a machine fusion architecture that separates control and estimation, goal-setting and goal-achieving behaviors. This model is called Frankel-Bedworth architecture [Frankel and Bedworth 2000] and is depicted in Figure 13.

The local and global processes have different objectives and, consequently, different roles. The local process tries to achieve the specified goals and maintain the specified standards. Thus the local process has the Estimator role, which is similar to the previous fusion models and includes the following tasks:

Sense. Raw information is gathered by the information sources.

Perceive. *Stimuli* retrieved by sensing are dealt according to their relevance (*focus*), and the Controller is informed which stimuli are being used (*awareness*).

Direct. Based on the comprehension of the perception (*semantics*) the Estimator can provide a feedback (*alert*) to the Controller. The disparity between current situation and desired situation is evaluated. Then, the Estimator is fed forward with *desires* that specify new goals and timetables.

Manage. Based on the *objectives*, the Controller is activated to define what is practical (*pragmatics*) so the Estimator can provide an appropriate *response*. Then, the Estimator provides a feedback to the Controller by reporting the expectations about the provided decision (*sensitivity*).

Effect. Selected decisions (*responses*) are applied and the control loop is closed by sensing the changes in the environment.

Global control process manages the goals and the performance of the system during the execution of the local process. Thus, the global process has the Controller role, which is responsible for controlling and managing the Estimator role and includes the following tasks:

Orient. The importance or relevance of sensed stimuli is configured.

Prefer. *Priority* is given to the aspects that are most relevant to the goal-achieving behavior, detailing the local goals (*desires*).

Expect. *Predictions* are made and the intentional objectives are filtered, determining what is practical to the Estimator *pragmatics*.

The Frankel-Bedworth architecture introduces the notion of a global process separated from the local process. The global control process rules the local process by controlling and defining its goals and monitoring its performance. On the other hand, the local process is supposed to implement and perform fusion methods and algorithms to accomplish the system's objectives. This architecture extends the previous models that were concerned only with the local process aspects. However, further discussions on how to effectively use this architecture to design and implement real information fusion systems are still desirable.

In real WSNs, the global control process will most likely be performed by human beings who feed the network with operation guidelines (*priorities*, *desires*, and *pragmatics*), whereas the local estimation process should be implemented within the computational system (sensor nodes and integrated systems). In this context, whenever necessary, the global process can feed the local process through interest dissemination by using, for instance, the Directed Diffusion communication paradigm [Intanagonwiwat et al. 2003] or a query interface such as the TinyDB [Madden et al. 2005].

For the sake of illustration, let us consider a combined application of environmental data gathering and target tracking. The *Sense* task is performed by sensor units that provide observations, which are selected by the *Perceive* task according to the focus. For instance, when a target is being detected, environmental data, such as temperature, may be discarded since trajectory information is more relevant in this case. During the *Direct* task, if the local process (Estimator) detects that the target is not a threat, it should alert the global process (Controller), which can ask again for low rate environmental data (new *desires*). Based on the new objective, the local process may change routes and notification rates in the *Manage* task, which is implemented by the sensor nodes (*Effect* task). Among other things, the global process can be used to: (i) specify an aggregation algorithm [Madden et al. 2002; Boulis et al. 2003a; Hellerstein et al. 2003] depending on current objectives (e.g., energy savings or data quality); (ii) reconfigure parameters of fusion algorithms such as the window size of moving average filters [Nakamura et al. 2005a] and the number of samples of particle filters [Sheng et al. 2005]; and (iii) select the best routing strategy based on the data generation profile [Figueiredo et al. 2004].

5.3.3. Some Remarks on Role-Based Models. Role-based models are fine-grained models that specify the actors and their roles in the fusion task. These models represent the evolved maturity level in the fusion domain. Although these models provide a better understanding of the fusion task, they do not explicitly consider the particularities of the WSNs (so do the information-based and activity-based models).

6. INFORMATION FUSION AND DATA COMMUNICATION

In WSNs, information fusion is closely related to data communication. The reason is that due to the limited power sources of current sensor nodes, it is usually desirable to take advantage of the limited computation capacity of sensor nodes to perform in-network fusion to reduce the overall data traffic (see Section 2.2). Thus, in this section we discuss some relevant aspects regarding the relationship between information fusion and data communication.

6.1. Distributed-Computing Paradigms

Different distributed computing paradigms have been adopted in WSNs, and depending on the chosen paradigm, information fusion occurs in different ways. In this section, we discuss the use of information fusion within different distributed computing paradigms, namely the In-Network Aggregation, Client-Server, Active Networks, and Mobile Agents paradigms.

6.1.1. In-Network Aggregation. The In-Network Aggregation is the most popular distributed-computing paradigm in WSNs. The idea is to take advantage of the node computation capacity and perform the desired fusion algorithm while data is routed towards the sink node. For that reason, this paradigm is also referred to as Data-Centric

Routing. Finding an optimal routing tree, connecting sources to sinks, is shown to be an NP-complete problem very similar to the Steiner Tree [Krishnamachari et al. 2002]. The Directed Diffusion solution [Intanagonwiwat et al. 2000; 2003] is the pioneer work for using the publish-subscribe approach with in-network data aggregation in WSNs. A key feature of Directed Diffusion is that events are sent and arrive asynchronously, and when they arrive at a node, they trigger callbacks to relevant applications (subscribers), thus performing the in-network data processing. These applications are the ones that implement and execute the desired information fusion algorithms. In this solution, the details of how published data are delivered to subscribers depend on the implementation of the so called filters, which are actually the routing and fusion algorithms.

Depending on the network organization, in-network aggregation may occur in different ways, according to the routing strategy. In flat networks, every node is functionally the same and data are routed in a multihop fashion since not every node directly reaches the sink. Thus, information fusion should be executed by every node that takes part in the routing process, and all fusion algorithms must be implemented by every node. Examples of multihop communication with in-network aggregation include the Directed Diffusion family of algorithms [Intanagonwiwat et al. 2003; Heidemann et al. 2003] and tree-based routing algorithms [Sohrabi et al. 2000; Krishnamachari et al. 2002; Zhou and Krishnamachari 2003]. In hierarchical networks, we usually have a two-hop communication. One hop for the cluster members to reach the cluster-head, and another hop for cluster-heads to reach the sink node. In this type of communication, information fusion is performed by cluster-heads that send the results to the sink. The first hierarchical solution for WSNs was the LEACH [Heinzelman et al. 2000], but several others have been proposed since then [Halgamuge et al. 2003; Kochhal et al. 2003; Mhatre and Rosenberg 2004; Hoang and Motani 2005a; Gupta et al. 2005]. In a hybrid solution, we might have multiple hops connecting source nodes to their cluster-head and/or multiple hops connecting cluster-heads to the sink. Thus, in such a scenario we may combine flat and hierarchical in-network aggregation. The strategy proposed by Nakamura et al. [2006] illustrates a routing algorithm for hybrid networks performing in-network data aggregation.

6.1.2. Client-Server. The traditional client-server model, as we have in the Internet, demands the knowledge, at every node, of the existence of the communicating nodes (servers) along with their addresses. In WSNs, however, we can relax this restriction into a data-centric approach wherein instead of knowing the nodes' addresses we need only know data names (e.g., temperature and movement). In this context, data-centric storage systems may be seen as a data-centric-client-server variant in the sense that, in such systems, data are stored by name into a node or a set of nodes (data servers), and when a user or another sensor node (data client) searches for a specific data, it may directly query the node storing that type of data [Ratnasamy et al. 2003; Li et al. 2003; Gummadi et al. 2005; Sheng et al. 2006; Ahn and Krishnamachari 2006]. In addition, instead of knowing nodes' addresses we only have to know data names. From the fusion perspective, nodes storing data may answer the query by performing the desired fusion algorithm (especially aggregation functions), and forwarding only the result. When the fusion algorithm requires different types of data from different data servers, we can combine the Client-Server and the In-Network Data Aggregation paradigms so that information fusion is also performed along the routing path. A less interesting approach of the client-server paradigm occurs when sensor nodes (fusion clients) send their data to the sink (fusion server) and data fusion is executed. Xu and Qi [2004] show that this last approach is interesting when we have only few sources and small-scale networks.

6.1.3. Active Networks. Active networks allow the injection of customized programs into the network nodes [Psounis 1999]. Accordingly, this paradigm allows high complexity and customizable computations to be performed within the network. In this case, information fusion may travel in the network as active packets, allowing different methods and applications (even unpredicted ones) to be executed at different moments, instead of storing every possible fusion algorithm into the nodes. Particularly, the Maté [Levis and Culler 2002] and the SensorWare [Boulis et al. 2003b] frameworks were proposed to implement the Active Networks paradigm into sensor networks. This paradigm is especially interesting for at least two scenarios: (i) when we cannot predict the application's behavior (e.g., an exploratory WSN deployed in Mars); and (ii) when we need to design long-lived networks whose applications may need to be remotely changed.

6.1.4. Mobile Agents. Mobile agents are programs that can migrate from node to node in a network, at times and to places of their own choosing. The state of the running program is saved, sent to the new node and restored, so the program can continue from the point it stopped [Kotz and Gray 1999]. Xu and Qi [2004] evaluate the use of mobile agents to perform information fusion in WSNs and show that, in contrast to the client-server model, this paradigm saves the network bandwidth and provides an effective way to overcome network latency when the number of nodes is large, which should often be the case. Similar conclusions were related by Qi et al. [2002] who took an example of target classification to describe the design and implementation of the mobile-agent-based distributed sensor network and illustrate the efficiency of the Mobile Agents paradigm. The order that nodes are visited in a WSN by the agent along the route affects the quality and cost of the fused data. In fact, computing a route for a mobile agent that fuses data as it visits nodes is NP-complete [Wu et al. 2004]. Wu et al. [2004] present an optimization formulation and a genetic algorithm for statically, off-line, finding the best route for an information fusion mobile agent executing a target tracking function.

6.2. Information Fusion and Data Communication Protocols

Regarding the relationship of information fusion and data communication protocols in WSNs, information fusion can play a supporting role or a leading role. In the former, we have information fusion acting as a tool to assist the communication protocol establishment, whereas in the latter, the communication protocols are designed to support an information fusion application (e.g., data aggregation target tracking). For the sake of exemplification, in this section we briefly revisit some references, putting them in the context of the role of information fusion.

6.2.1. Information Fusion as a Supporting Role. Information fusion is a promising tool to support different tasks in WSNs. All tasks demanding any sort of parameter estimation can benefit from the methods discussed in Section 4.2. Similarly, every inference-based decision may use the techniques presented in Section 4.1. Currently, information fusion has started to be used as a supporting role to assist communication protocols but its potential is far from being fully explored.

MAC protocols have used information fusion techniques intensively. Fuzzy logic is used by Wallace et al. [2005] and Liang and Ren [2005b] to define nodes' duty cycle in the MAC layer. Wallace et al. [2005] propose a fuzzy-based approach that—based on nodes' transmit-queue size, residual energy, and collision rate—defines the nodes' duty cycle so that nodes with high transmit queue have priority to access the medium. Moving average filters have been used by MAC protocols with different purposes such as: estimating ambient noise to determine whether the channel is clear [Polastre et al. 2004]; local clock synchronization for contention purposes [Rhee et al. 2005]; and detecting incipient congestion for fair and efficient rate control [Rangwala et al. 2006]. Kalman filters have been used to predict the frame size, avoiding the transmission of large frames whenever possible [Ci et al. 2004; Raviraj et al. 2005; Ci and Sharif 2005].

We can also point out some routing solutions that use information fusion searching for improved performance. Fuzzy logic has been used to decide the nodes participating in the routing path [Liang and Ren 2005a; Srinivasan et al. 2006]. In order to improve the network lifetime, Liang and Ren [2005a] use fuzzy logic to evaluate different parameters—such as battery capacity, mobility, and distance to the destination—and choose the nodes to be included in the routing path. Woo et al. [2003] use moving average filters within adaptive link estimators so that link connectivity statistics are exploited by routing protocols to reduce packet losses. Nakamura et al. [2005b] use the moving average filter to estimate the data traffic of continuous WSNs, and that estimate is further used to detect routing failure by means of the Dempster-Shafer inference. The SCAR algorithm [Mascolo and Musolesi 2006] uses the Kalman filter to predict context information (mobility and resources) about its neighbors, and choose the best neighbor for routing its data. Hartl and Li [2004] use maximum likelihood to estimate per-node loss rates during the aggregation and reporting of data from sources to sink nodes, which can be used to bypass lossy areas.

Localized algorithms, wherein nodes make decisions based on neighbors' information (e.g., link quality, residual energy, connectivity, and mobility), can take advantage of dual prediction schemes to reduce communication. In this scheme, two neighbor nodes simultaneously apply a predictive estimator (e.g., the Kalman filter) so that a node only exchanges data when it knows its parameters are not being correctly predicted by its neighbor. Furthermore, besides using information fusion methods to estimate parameters, such as residual energy, inference techniques can also be used to make decisions. For instance, MAC protocols may use the Bayesian inference or neural networks to accurately decide whether or not it is worth trying to transmit data given the current link quality, resources, and QoS requirements. To determine whether or not the applicability of fusion methods in such situations is feasible, we must evaluate the computational cost of the fusion algorithms, the resultant delay, the energy consumed, and the impact on the quality of the service provided by communication protocol.

6.2.2. Information Fusion as a Leading Role. In many cases, we cannot distinguish information fusion algorithms from the application, in the sense that, to accomplish the application objectives we execute one or multiple information fusion algorithms. For instance, target tracking is essentially the application of information fusion algorithms such as Kalman, or particle filters; an event detection is essentially an inference task that may use an information fusion technique such as the Bayesian or Dempster-Shafer inference. When information fusion plays such a leading role (application) in the network, the way communication is established may affect the results regarding data quality and energy consumption. In the following, we discuss a few solutions that were proposed to optimize the performance of specific information fusion applications.

Having in mind the different applications, the Directed Diffusion paradigm [Intanagonwiwat et al. 2003] provides a communication framework wherein virtually any information fusion application can be implemented by using filters. These filters are special components responsible for guiding the routing process and the fusion processes simultaneously. In order to improve the detection ability of a WSN, Kochhal et al. [2003] propose a role-based clustering algorithm, which considers the sensing ability of the nodes, and organizes the network by recursively finding connected dominating sets. Those sets are used to define coordinators (cluster-heads), and routing nodes; the remaining nodes become sensing collaborators (sources).

In data aggregation applications, a sink node is interested in collecting aggregated data from a subset of nodes. In this context, data communication should use as few nodes and resources as possible to ensure the delivery and aggregation of data generated by source nodes. This is essentially an NP-complete problem similar to the Steiner tree; some heuristics have been proposed for that problem. Three heuristics are evaluated by Krishnamachari et al. [2002]: the centered-at-nearest-source tree (CNS), the shortest-path tree (SPT), and the greedy incremental tree (GIT). In the CNS, each source sends its data directly to the source closest to the sink; in the SPT, each source sends its data to the sink along the shortest path between both nodes; and in the GIT, the routing tree starts with the shortest path between the sink and the nearest source, and at each step after that, the source closest to the current tree is included in the tree. As Krishnamachari et al. [2002] show, the GIT heuristic is the best of the three. However, its distributed version [Bauer and Varma 1996] demands a lot of communication and memory usage, because every node needs to know its shortest paths to the other nodes in the network. Motivated by that infeasible cost, Nakamura et al. [2006] propose the InFRA heuristic, which finds the shortest paths that maximize data aggregation, and has an $O(1)$-approximation ratio. Zhu et al. [2005] present a heuristic, called Semantic/Spatial Correlation-aware Tree (SCT), that is constructed during the course of a query delivery. The SCT builds a fixed aggregation backbone that simplifies the generation of efficient aggregation trees, and is independent of source distribution and density. However, in contrast to the InFRA heuristics [Nakamura et al. 2006], the SCT needs to be pro-actively rebuilt, leading to energy waste. For the same problem, Ding et al. [2003] propose a tree-based routing algorithm based on nodes' residual energy, so that nodes with more energy are likely to perform data aggregation and routing. Once the tree is built, leaf nodes are turned off to save energy, but no approximation ratio is provided for this heuristic.

Another approach to the aforementioned problem is the use of role assignment algorithms to define which nodes are to be used and what actions those nodes should take. Bhardwaj and Chandakasan [2002] derive upper bounds on the lifetime of WSNs that perform information fusion by assigning roles (sensor, relay, and aggregator), and modelling the optimal role assignment as a linear problem to find the assignment that maximizes the network lifetime. By computing a user-defined cost function, Bonfils and Bonnet [2003] propose an adaptive and decentralized solution that progressively refines the role assignment. The SPRING algorithm [Dasgupta et al. 2003] for mobile sensor networks defines two roles (sensor and relay/aggregator), and places nodes and assigns roles to them so the system's lifetime is maximized and the region of interest is covered by at least one sensor node. In the DFuse framework [Kumar et al. 2003], role assignment is provided by a heuristic in which a tree with a naive role assignment is created, then nodes exchange health information, and the role is transferred to the neighbor with the best health regarding a given cost function. Frank and Römer [2005] propose a basic structure for a generic role assignment framework with applications for coverage, clustering, and in-network aggregation. Similarly to the filter approach of Directed Diffusion, the network designer should specify roles and assignment rules.

When we have information fusion as a leading role, source selection and route selections are problems of major concern. Taking target tracking applications based on particle filters as an example, selecting good particles (samples) for estimating a target's trajectory is challenging because the fewer particles the cheaper the computation. In this context, Zhao et al. [2002a; 2003a] propose an information-directed approach in which sources and communicating nodes are chosen by dynamically optimizing the information utility of data for a given cost of communication and computation.

Chen et al. [2006c] propose the Energy-Efficient Protocol for Aggregator Selection (EPAS) for selecting nodes that perform information fusion. The authors derive the optimal number of aggregators, and present fully distributed algorithms for the aggregator selection. A key contribution is that these algorithms are independent of routing protocols. Chen et al. [2006b] use a cluster-based communication architecture, based on LEACH [Heinzelman et al. 2000], wherein data aggregation runs parallel to the cluster-heads, improving the energy efficiency via meta data negotiation. In addition, for each event and each cluster, only one of the cluster members is selected to send data to the cluster-head. Zhou et al. [2004] use Directed Diffusion to provide a hierarchical aggregation scheme for WSNs to improve reliability and provide more applicable data aggregation.

7. FINAL REMARKS

This article presents the background necessary to answer some questions about information fusion, such as: (1) What is information fusion? (2) Why should a designer use it? (3) What are the available techniques? and (4) How should a designer use such techniques? In simple words, the answers are:

(1) Information fusion is the set of resources used to combine multiple sources such that the result is in some sense better than the individual inputs.
(2) Information fusion should be used to improve the performance of a task by understanding the current situation, and supporting decisions.
(3) The techniques include filters, Bayesian and Dempster-Shafer inference, aggregation functions, interval combination functions, and classification methods.
(4) The use of the fusion techniques should be guided by architectures and models, such as the JDL model.

The provided background supports the design of fusion-based solutions for different levels of applications in a WSN, such as internal tasks (e.g., data routing) and system applications (e.g., target detection). However, there are some limitations regarding the methods and the architectures that should be considered.

Depending on the model adopted, some of the listed methods might be too expensive to be executed by current sensor nodes. For example, in the Dempster-Shafer inference, the combination rule has exponential cost regarding the number of states in the frame of discernment. Thus, if two logically different states are functionally the same, from the application perspective, then they should be modelled as a single state for the sake of performance.

Other methods might be improved to operate in a distributed fashion. One of the great challenges is to assure temporal and spatial correlation among the sources while the data is fused and disseminated at the same time.

Current fusion architectures are weak in considering the peculiarities of WSNs because they are not network-driven. However, we understand that such architectures may be applied within specific models for WSNs, wherein the whole network is designed based on a global architecture for WSNs; then, the fusion task can be designed based on a fusion model, respecting the requirements established by the global architecture.

REFERENCES

ABDELBAR, A. M., ANDREWS, E. A. M., AND WUNSCH III, D. C. 2003. Abductive reasoning with recurrent neural networks. *Neural Netw. 16,* 5-6, 665–673.

AGUERO, J. R. AND VARGAS, A. 2005. Inference of operative configuration of distribution networks using fuzzy logic techniques—part II: Extended real-time model. *IEEE Trans. Power Syst. 20,* 3 (August), 1562–1569.

AHLSWEDE, R., CAI, N., LI, S. R., AND YEUNG, R. W. 2000. Network information flow. *IEEE Trans. Inf. Theory 46,* 4 (July), 1204–1216.

AHMED, M. AND POTTIE, G. 2005. Fusion in the context of information theory. In *Distributed Sensor Networks*, S. S. Iyengar and R. R. Brooks, Eds. CRC Press, Boca Raton, Chapter 22, 419–436.

AHN, J. AND KRISHNAMACHARI, B. 2006. Fundamental scaling laws for energy-efficient storage and querying in wireless sensor networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*. ACM, Florence, Italy, 334–343.

AKYILDIZ, I. F., ESTRIN, D., CULLER, D. E., AND SRIVASTAVA, M. B., Eds. 2003. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*. ACM, Los Angeles.

AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CYIRCI, E. 2002. Wireless sensor networks: A survey. *Comput. Netw. 38,* 4 (March), 393–422.

ARBUCKLE, D., HOWARD, A., AND MATARIC, M. J. 2002. Temporal occupancy grids: A method for classifying spatio-temporal properties of the environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Switzerland, 409–414.

ARULAMPALAM, M. S., MASKELL, S., GORDON, N., AND CLAPP, T. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Sig. Proc. 50,* 2 (February), 174–188. IEEE.

ASLAM, J., BUTLER, Z., CONSTANTIN, F., CRESPI, V., CYBENKO, G., AND RUS, D. 2003. Tracking a moving object with a binary sensor network. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, 150–161.

BAJWA, W., HAUPT, J., SAYEED, A., AND NOWAK, R. 2006. Compressive wireless sensing. *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 134–142.

BANON, G. 1981. Distinction between several subsets of fuzzy measures. *Fuzzy Sets Syst. 5,* 3 (May), 291–305.

BARAN, R. H. 1989. A collective computation approach to automatic target recognition. In *Proceedings of the International Joint Conference on Neural Networks*. Vol. I. IEEE, Washington, D.C., 39–44.

BARROS, J. AND SERVETTO, S. D. 2006. Network information flow with correlated sources. *IEEE Trans. Inf. Theory 52,* 1 (January), 155–170.

BASS, T. 2000. Intrusion detection systems and multisensor data fusion. *Comm. ACM 43,* 4 (April), 99–105. ACM Press.

BAUER, F. AND VARMA, A. 1996. Distributed algorithms for multicast path setup in data networks. *Trans. Netw. 4,* 2 (April), 181–191.

BAYES, T. R. 1763. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society 53*, 370–418.

BEDWORTH, M. D. AND O'BRIEN, J. C. 1999. The omnibus model: A new model for data fusion? In *Proceedings of the 2nd International Conference on Information Fusion (FUSION'99)*. ISIF, Sunnyvale, 437–444.

BEZDEK, J. C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. Kluwer Academic Publishers, Norwell, MA.

BHARDWAJ, M. AND CHANDAKASAN, A. P. 2002. Bounding the lifetime of sensor networks via optimal role assignment. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*. USC/Information Sciences Institute, IEEE, New York, NY.

BISWAS, R., THRUN, S., AND GUIBAS, L. J. 2004. A probabilistic approach to inference with limited information in sensor networks. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 269–276.

BLATT, D. AND HERO, A. 2004. Distributed maximum likelihood estimation for sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*. Vol. 3. IEEE, Montreal, Canada, 929–932.

BLUM, P., MEIER, L., AND THIELE, L. 2004. Improved interval-based clock synchronization in sensor networks. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 349–358.

BLUMENTHAL, J., TIMMERMANN, D., BUSCHMANN, C., FISCHER, S., KOBERSTEIN, J., AND LUTTENBERGER, N. 2006. Minimal transmission power as distance estimation for precise localization in sensor networks. In *Proceedings of the 2006 International Conference on Communications and Mobile Computing (IWCMC'06)*. ACM, Vancouver, British Columbia, Canada, 1331–1336.

BONFILS, B. J. AND BONNET, P. 2003. Adaptive and decentralized operator placement for in-network query processing. In *Information Processing in Sensor Networks 2nd International Workshop (IPSN'03)*, 47–62.

BONISSONE, P. P. 1997. Soft computing: The convergence of emerging reasoning technologies. *Soft Comput. 1,* 1 (April), 6–18.

BOULIS, A., GANERIWAL, S., AND SRIVASTAVA, M. B. 2003a. Aggregation in sensor networks: An energy-accuracy trade-off. *Ad Hoc Networks 1,* 2-3 (September), 317–331. Special Issue on Sensor Network Protocols and Applications.

BOULIS, A., HAN, C.-C., AND SRIVASTAVA, M. B. 2003b. Design and implementation of a framework for efficient and programmable sensor networks. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys'03)*. USENIX, San Francisco, CA, 187–200.

BOYD, J. R. 1987. A discourse on winning and losing. Unpublished set of briefing slides available at Air University Library, Maxwell AFB, Alabama.

BRACIO, B. R., HORN, W., AND MÖLLER, D. P. F. 1997. Sensor fusion in biomedical systems. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 3. IEEE, Chicago, IL, 1387–1390.

BROKMANN, G., MARCH, B., ROMHILD, D., AND STEINKE, A. 2001. Integrated multisensors for industrial humidity measurement. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, Baden-Baden, Germany, 201–203.

BROOKS, R. R. AND IYENGAR, S. 1998. *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall PTR, Upper Saddle River, NJ.

BROWN, C., DURRANT-WHYTE, H., LEONARD, J., RAO, B., AND STEER, B. 1992. Distributed data fusion using Kalman filtering: A robotics application. In *Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzalez, Eds. Academic Press, Inc., San Diego, CA, Chapter 7, 267–309.

BROWN, R. G. AND HWANG, P. Y. C. 1996. *Introduction to Random Signals and Applied Kalman Filtering*, 3rd ed. John Wiley & Sons, New York, NY.

BUEDE, D. M. 1988. Shafer-Dempster and Bayesian reasoning: A response to "Shafer-Dempster reasoning with applications to multisensor target identification systems." *IEEE Trans. Syst., Man Cyber. 18,* 6 (November/December), 1009–1011.

CAIN, M. P., STEWART, S. A., AND MORSE, J. B. 1989. Object classification using multispectral sensor data fusion. In *Proceedings of SPIE Sensor Fusion II*. Vol. 1100. SPIE, Orlando, FL, 53–61.

CASTELAZ, P. F. 1988. Neural networks in defense applications. In *Proceedings of the IEEE International Conference on Neural Networks*. Vol. II. IEEE, San Diego, CA, 473–480.

CHAKRABARTY, K., IYENGAR, S. S., QI, H., AND CHO, E. 2002. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Comput. 51,* 12 (December), 1448–1453.

CHAKRAVARTY, P. AND JARVIS, R. 2005. Multiple target tracking for surveillance: A particle filter approach. In *Proceedings of the 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP'05)*. IEEE, Melbourne, Australia, 181–186.

CHAN YET, W. AND QIDWAI, U. 2005. Intelligent sensor network for obstacle avoidance strategy. In *Proceedings of the 4th IEEE Conference on Sensors*. IEEE, Irvine.

CHEN, C., ALI, A. M., AND WANG, H. 2006a. Design and testing of robust acoustic arrays for localization and enhancement of several bird sources. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 268–275.

CHEN, H., MINENO, H., AND MIZUNO, T. 2006b. A meta-data-based data aggregation scheme in clustering wireless sensor networks. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*. IEEE, Nara, Japan, 154–154.

CHEN, Y. P., LIESTMAN, A. L., AND LIU, J. 2006c. A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. *IEEE Trans. Vehic. Tech. 55,* 3 (May), 789–796.

CHENG, Y. AND KASHYAP, R. L. 1988. Comparison of Bayesian and Dempster's rules in evidence combination. In *Maximum-Entropy and Bayesian Methods in Science and Engineering*, G. J. Erickson and C. R. Smith, Eds. Klewer, Dordrecht, Netherlands, 427–433.

CHEW, P. AND MARZULLO, K. 1991. Masking failures of multidimentional sensors. In *Proceedings of the 10th Symposium on Reliable Distributed Systems*. IEEE, Pisa, Italy, 32–41.

CI, S. AND SHARIF, H. 2005. Performance comparison of kalman filter based approaches for energy efficiency in wireless sensor networks. In *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'05)*. IEEE, Cairo, Egypt, 58–65.

CI, S., SHARIF, H., AND NULI, K. 2004. A UKF-based link adaptation scheme to enhance energy efficiency in wireless sensor networks. In *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'04)*. Vol. 4. IEEE, Barcelona, Spain, 2483–2488.

CIANCIO, A. AND ORTEGA, A. 2004. A distributed wavelet compression algorithm for wireless sensor networks using lifting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*. Vol. 4. IEEE, Montreal, Canada.

CIANCIO, A., PATTEM, S., ORTEGA, A., AND KRISHNAMACHARI, B. 2006. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 309–316.

CIMANDER, C., CARLSSON, M., AND MANDENIUS, C. 2002. Sensor fusion for on-line monitoring of yoghurt fermentation. *J. Biotech. 99,* 3 (November), 237–248.

COATES, M. 2004. Distributed particle filters for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 99–107.

COHEN, N. H., PURAKAYASTHA, A., TUREK, J., WONG, L., AND YEH, D. 2001. Challenges in flexible aggregation of pervasive data. IBM Research Report RC 21942 (98646), IBM Research Division, Yorktown Heights, NY (January).

COUÉ, C., FRAICHARD, T., BESSIERE, P., AND MAZER, E. 2002. Multi-sensor data fusion using Bayesian programming: An automotive application. In *IEEE/RSJ International Conference on Intelligent Robots and System*. Vol. 1. IEEE, Lausanne, Switzerland, 141–146.

CRISAN, D. AND DOUCET, A. 2002. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Sig. Proc. 50,* 3 (March), 736–746.

CUI, X., HARDIN, T., RAGADE, R., AND ELMAGHRABY, A. 2004. A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*. IEEE, Fort Lauderdale, 194–203.

DASARATHY, B. V. 1997. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proc. IEEE 85,* 1 (January), 24–38.

DASARATHY, B. V. 2000. More the merrier... or is it?—sensor suite augmentation benefits assessment. In *Proceedings of the 3rd International Conference on Information Fusion (Fusion 2000)*. Vol. 2. IEEE, Paris, France, WEC3/20–WEC3/25.

DASARATHY, B. V. 2001. What, where, why, when, and how? *Inform. Fus. 2,* 2 (January), 75–76. Editorial.

DASGUPTA, K., KUKREJA, M., AND KALPAKIS, K. 2003. Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. In *Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC'03)*. Vol. 1. IEEE, Antalya, Turkey, 341–348.

DATA FUSION SERVER. 2004. [Online] Available: http://www.data-fusion.org.

DE CAMPOS, L. M., GAMEZ, J. A., AND MORAL, S. 2002. Partial abductive inference in Bayesian belief networks—an evolutionary computation approach by using problem-specific genetic operators. *IEEE Trans. Evolut. Computat. 6,* 2 (April), 105–131.

DEMPSTER, A. P. 1968. A generalization of Bayesian inference. *J. Royal Stat. Soc., Series B 30*, 205–247.

DHILLON, S. S., CHAKRABARTY, K., AND IYENGAR, S. S. 2002. Sensor placement for grid coverage under imprecise detections. In *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*. Vol. 2. IEEE, Annapolis, Maryland, 1581–1587.

DING, M., CHENG, X., AND XUE, G. 2003. Aggregation tree construction in sensor networks. In *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC-Fall 2003)*. Vol. 4. IEEE, Orlando, 2168–2172.

DOUCET, A., VO, B.-N., ANDRIEU, C., AND DAVY, M. 2002. Particle filtering for multi-target tracking and sensor management. In *Proceedings of the 5th International Conference on Information Fusion (Fusion 2002)*. Vol. 1. IEEE, Annapolis, Maryland, 474–481.

DUARTE, M. F., WAKIN, M. B., BARON, D., AND BARANIUK, R. G. 2006. Universal distributed sensing via random projections. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 177–185.

DURRANT-WHYTE, H. F. 1988. Sensor models and multisensor integration. *Inter. J. Robotics Res. 7,* 6 (December), 97–113.

ELFES, A. 1987. Sonar-based real-world mapping and navigation. *IEEE J Robotics Automat. RA-3,* 3 (June), 249–265.

ELFES, A. 1989. Using occupancy grids for mobile robot perception and navigation. *IEEE Comput. 22,* 6 (June), 46–57.

ELMENREICH, W. 2002. Sensor fusion in time-triggered systems. Ph.D. thesis, Institut für Technische Informatik, Vienna University of Technology, Vienna, Austria.

ELSON, J. AND PARKER, A. 2006. Tinker: A tool for designing data-centric sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 350–357.

FANG, L., DU, W., AND NING, P. 2005. A beacon-less location discovery scheme for wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 161–171.

FIGUEIREDO, C. M., NAKAMURA, E. F., AND LOUREIRO, A. A. 2004. Multi: A hybrid adaptive dissemination protocol for wireless sensor networks. In *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)*. Lecture Notes in Computer Science, vol. 3121. Springer, Turku, Finland, 171–186.

Filippidis, A., Jain, L. C., and Martin, N. 2000. Fusion of intelligent agents for the detection of aircraft in SAR images. *IEEE Trans. Patt. Anal. Mach. Intell. 22,* 4 (April), 378–384.

Forney, G. D. 1973. The viterbi algorithm. *Proc. IEEE 61,* 3 (March), 268–278.

Fowler, M. L. and Chen, M. 2005. Fisher-information-based data compression for estimation using two sensors. *IEEE Trans. Aerospace Electr. Syst. 41,* 3 (July), 1131–1137.

Frank, C. and Römer, K. 2005. Algorithms for generic role assignment in wireless sensor networks. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*, 230–242.

Frankel, C. B. 1999. Such order from confusion sprung: Adaptive competence and affect regulation. Ph.D. thesis, Pacific Graduate School of Psychology, Palo Alto, CA.

Frankel, C. B. and Bedworth, M. D. 2000. Control, estimation and abstraction in fusion architectures: Lessons from human information processing. In *Proceedings of the 3rd International Conference on Information Fusion (Fusion 2000)*. Vol. 1. IEEE, Paris, France, MoC5/3–MoC5/10.

Friedlander, D. S. 2005. Semantic information extraction. In *Distributed Sensor Networks*, S. S. Iyengar and R. R. Brooks, Eds. CRC Press, Boca Raton, Chapter 21, 409–417.

Friedlander, D. S. and Phoha, S. 2002. Semantic information fusion for coordinated signal processing in mobile sensor networks. *Int. J. High Perf. Comput. Appl. 16,* 3 (Fall), 235–241.

Gagvani, N. and Silver, D. 2000. Shape-based volumetric collision detection. In *Proceedings of the 2000 IEEE Symposium on Volume Visualization*. IEEE, Salt Lake City, Utah, 57–61.

Gallager, R. G. 1968. *Information Theory and Reliable Communication*. Wiley, New York.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. 2003. Timing-sync protocol for sensor networks. In *Proceedings of the 1st International Conference on Embedded Networks Sensor Systems (SenSys'03)*, 138–149.

Gao, J. B. and Harris, C. J. 2002. Some remarks on Kalman filters for the multisensor fusion. *Information Fusion 3,* 3 (September), 191–201.

Garvey, T. D., Lowrance, J. D., and Fischler, M. A. 1981. An inference technique for integrating knowledge from disparate sources. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI'81)*. William Kaufmann, Vancouver, British Columbia, Canada, 319–325.

Geoscience and Remote Sensing Society. 2004. [Online] Available: http://www.dfc-grss.org.

Gilks, W., Richardson, S., and Spie, D., Eds. 1996. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, London, UK.

Grime, S. and Durrant-Whyte, H. F. 1994. Data fusion in decentralized sensor networks. *Contr. Eng. Pract. 2,* 5 (October), 849–863.

Gu, L., Jia, D., Vicaire, P., Yan, T., Luo, L., Tirumala, A., Cao, Q., He, T., Stankovic, J. A., Abdelzaher, T., and Krogh, B. H. 2005. Lightweight detection and classification for wireless sensor networks in realistic environments. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*, 205–217.

Guestrin, C., Bodik, P., Thibaux, R., Paskin, M., and Madden, S. 2004. Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 1–10.

Gummadi, R., Li, X., Govindan, R., Shahabi, C., and Hong, W. 2005. Energy-efficient data organization and query processing in sensor networks. *SIGBED Review 2,* 1, 7–12.

Gunnarsson, F. and Gustafsson, F. 2003. Positioning using time-difference of arrival measurements. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 6. IEEE, Hong Kong, 553–556.

Guo, D. and Wang, X. 2004. Dynamic sensor collaboration via sequential Monte Carlo. *IEEE J. Selec. Areas Comm. 22,* 6 (August), 1037–1047.

Guo, D., Wang, X., and Chen, R. 2005. New sequential Monte Carlo methods for nonlinear dynamic systems. *Statis. Comput. 15,* 2 (April), 135–147.

Gupta, I., Riordan, D., and Sampalli, S. 2005. Cluster-head election using fuzzy logic for wireless sensor networks. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR'05)*. IEEE, Halifax, Canada, 255–260.

Halgamuge, M. N., Guru, S. M., and Jennings, A. 2003. Energy efficient cluster formation in wireless sensor networks. In *Proceedings of the 10th International Conference on Telecommunications (ICT'03)*. Vol. 2. IEEE, Papeete, French Polynesia, 1571–1576.

Hall, D. L. 1992. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Norwood, MA.

Hall, D. L. and Llinas, J. 1997. An introduction to multi-sensor data fusion. *Proc. IEEE 85,* 1 (January), 6–23.

Hartl, G. and Li, B. 2004. Loss inference in wireless sensor networks based on data aggregation. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 396–404.

Hartl, G. and Li, B. 2005. infer: A Bayesian inference approach towards energy efficient data collection in dense sensor networks. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. IEEE, Washington, 371–380.

Haupt, J. and Nowak, R. 2006. Signal reconstruction from noisy random projections. *IEEE Trans. Inform. Theory 52,* 9 (September), 4036–4048.

He, T., Blum, B. M., Stankovic, J. A., and Abdelzaher, T. 2004. AIDA: Adaptive application independent data aggregation in wireless sensor network. *ACM Trans. Embed. Comput. Syst. 3,* 2 (May), 426–457. Special issue on Dynamically Adaptable Embedded Systems.

Heidemann, J., Silva, F., and Estrin, D. 2003. Matching data dissemination algorithms to application requirements. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, 218–229.

Heinzelman, W., Chandrakasan, A., and Balakrishnan, H. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS'00)*. IEEE, Maui, 8020–8029.

Hellerstein, J. M., Hong, W., Madden, S., and Stanek, K. 2003. Beyond average: Towards sophisticated sensing with queries. In *Information Processing in Sensor Networks: 2nd International Workshop (IPSN'03)*, 63–79.

Hoang, A. T. and Motani, M. 2005a. Collaborative broadcasting and compression in cluster-based wireless sensor networks. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks (EWSN'05)*. IEEE, Istanbul, Turkey, 197–206.

Hoang, A. T. and Motani, M. 2005b. Exploiting wireless broadcast in spatially correlated sensor networks. In *Proceedings of the 2005 IEEE International Conference on Communications (ICC'05)*. Vol. 4. IEEE, Seoul, Korea, 2807–2811.

Hongyang, C., Ping, D., Yongjun, X., and Xiaowei, L. 2005. A robust location algorithm with biased extended Kalman filtering of TDOA data for wireless sensor networks. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WCNM'05)*. Vol. 2. IEEE, Wuhan, China, 883–886.

Hoover, A. and Olsen, B. D. 1999. A real-time occupancy map from multiple video streams. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 3. IEEE, Detroit, Michigan, 2261–2266.

Hoover, A. and Olsen, B. D. 2000. Sensor network perception for mobile robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1. IEEE, San Fransico, California, 342–347.

Hu, L. and Evans, D. 2004. Localization for mobile sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04)*. ACM, Philadelphia, PA, USA, 45–57.

Hua, G. and Chen, C. W. 2005. Distributed source coding in wireless sensor networks. In *Proceedings of the 2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (Qshine'05)*. IEEE, Orlando.

Ihler, A. T., Fisher I, J. W., Moses, R. L., and Willsky, A. S. 2005. Nonparametric belief propagation for self-localization of sensor networks. *IEEE J. Select. Areas Comm. 23,* 4 (April), 809–819.

INFOCOM, Ed. 2005. *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*. IEEE, Miami, USA.

Intanagonwiwat, C., Estrin, D., Govindan, R., and Heidemann, J. 2002. Impact of network density on data aggregation in wireless sensor networks. In *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS'02)*. IEEE, Vienna, Austria, 457–458.

Intanagonwiwat, C., Govindan, R., and Estrin, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*. ACM Press, Boston, MA, 56–67.

Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., and Silva, F. 2003. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw. 11,* 1 (February), 2–16.

International Society of Information Fusion. 2004. [Online] Available: http://www.inforfusion.org.

IPSN, Ed. 2005. *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*. IEEE, Los Angeles.

ISARD, M. AND BLAKE, A. 1996. Contour tracking by stochastic propagation of conditional density. In *4th European Conference on Computer Vision (ECCV'96)*, B. F. Buxton and R. Cipolla, Eds. Lecture Notes in Computer Science, vol. 1064. Springer, Cambridge, UK, 343–356.

ISLA, D. A. AND BLUMBERG, B. M. 2002. Object persistence for synthetic creatures. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, Bologna, Italy, 1356–1363.

IYENGAR, S. S., CHAKRABARTY, K., AND QI, H. 2001. Introduction to special issue on "distributed sensor networks for real-time systems with adaptive configuration." *J. Franklin Inst. 338,* 6 (September), 651–653.

JACOBS, O. L. R. 1993. *Introduction to Control Theory*, 2nd ed. Oxford University Press, Oxford, UK.

JAIN, A., CHANG, E. Y., AND WANG, Y.-F. 2004. Adaptive stream resource management using Kalman filters. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04)*. ACM, Paris, France, 11–22.

JAYASIMHA, D. N. 1994. Fault tolerance in a multisensor environment. In *Proceedings of the 13th Symposium on Reliable Distributed Systems*. IEEE, Dana Point, CA, 2–11.

JAZWINSKI, A. H. 1970. *Stochastic Processes and Filtering Theory*. Academic Press, New York.

JIN, G. AND NITTEL, S. 2006. Ned: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*. IEEE, Washington, DC, 153–161.

JU, H. AND CUI, L. 2005. Easipc: A packet compression mechanism for embedded WSN. In *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*. IEEE, Hong Kong, China, 394–399.

JULIER, S. J. AND UHLMANN, J. K. 1997. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*. Vol. 3068. SPIE, San Diego, 182–193.

KALMAN, R. E. 1960. A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Engin. 82*, 35–45.

KALPAKIS, K., DASGUPTA, K., AND NAMJOSHI, P. 2003. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Comput. Netw. 42,* 6 (August), 697–716.

KEPPENS, J., SHEN, Q., AND SCHAFER, B. 2005. Probabilistic abductive computation of evidence collection strategies in crime investigation. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law (ICAIL'05)*. ACM, Bologna, Italy, 215–224.

KESSLER ET AL., 1992. Functional description of the data fusion process. Tech. rep., Naval Air Development Center, Warminster, PA, USA. January. Report prepared for the Office of Naval Technology.

KLEIN, L. A. 1993. *Sensor and Data Fusion Concepts and Applications*. Vol. TT14. SPIE Optical Engineering Press.

KOCHHAL, M., SCHWIEBERT, L., AND GUPTA, S. 2003. Role-based hierarchical self organization for wireless ad hoc sensor networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA'03)*. ACM, San Diego, CA, 98–107.

KOHONEN, T. 1997. *Self-Organizing Maps*. Springer-Verlag, Secaucus, NJ, USA.

KOKAR, M. M., BEDWORTH, M. D., AND FRANKEL, C. B. 2000. A reference model for data fusion systems. In *Sensor Fusion: Architectures, Algorithms and Applications IV*. SPIE, Orlando, FL, 191–202.

KOKAR, M. M., TOMASIK, J. A., AND WEYMAN, J. 1999. A formal approach to information fusion. In *Proceedings of the 2nd International Conference on Information Fusion (Fusion'99)*. Vol. 1. ISIF, Sunnyvale, 133–140.

KOTZ, D. AND GRAY, R. S. 1999. Mobile agents and the future of the internet. *ACM SIGOPS Oper. Syst. Rev. 33,* 3 (July), 7–13.

KREUCHER, C., KASTELLA, K., AND HERO III, A. O. 2005. Sensor management using an active sensing approach. *Sig. Proc. 85,* 3 (March), 607–624.

KRISHNAMACHARI, B., ESTRIN, D., AND WICKER, S. 2002. The impact of data aggregation in wireless sensor networks. In *International Workshop of Distributed Event Based Systems (DEBS)*. IEEE, Vienna, Austria, 575–578.

KRISHNAMACHARI, B. AND IYENGAR, S. 2004. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. Comput. 53,* 3 (March), 241–250.

KULIK, J., HEINZELMAN, W., AND BALAKRISHNAN, H. 2002. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks 8,* 2/3 (March-May), 169–185.

KUMAR, R., WOLENETZ, M., AGARWALLA, B., SHIN, J., HUTTO, P., PAUL, A., AND RAMACHANDRAN, U. 2003. Dfuse: A framework for distributed data fusion. In *Proceedings of the 1st International Conference on Embedded Networks Sensor Systems (SenSys'03)*, 114–125.

Kumar, V. P. and Desai, U. B. 1996. Image interpretation using Bayesian networks. *IEEE Trans. Patt. Anal. Mach. Intell. 18,* 1 (January), 74–77.

Kusuma, J., Doherty, L., and Ramchandran, K. 2001. Distributed compression for sensor networks. In *Proceedings of the 2001 International Conference on Image Processing (ICIP'01)*. Vol. 1. IEEE, Thessaloniki, Greece, 82–85.

Lee, C. C. 1990. Fuzzy logic in control systems: Fuzzy logic controller—part i. *IEEE Trans. Syst., Man Cyber. 20,* 2 (March–April), 404–418.

Levis, P. and Culler, D. 2002. Maté: A tiny virtual machine for sensor networks. In *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. ACM Press, San Jose, CA, 85–95.

Lewis, T. W. and Powers, D. M. W. 2002. Audio-visual speech recognition using red exclusion and neural networks. In *Proceedings of the 25th Australasian Conference on Computer Science*. Australian Computer Society, Inc., Melbourne, Victoria, Australia, 149–156.

Li, S., Lin, Y., Son, S. H., Stankovic, J. A., and Wei, Y. 2004. Event detection services using data service middleware in distributed sensor networks. *Telecomm. Syst. 26,* 2–4 (June), 351–368.

Li, T., Ekpenyong, A., and Huang, Y. 2006. Source localization and tracking using distributed asynchronous sensor. *IEEE Trans. Sig. Proc. 54,* 10 (October), 3991–4003.

Li, X., Kim, Y. J., Govindan, R., and Hong, W. 2003. Multi-dimensional range queries in sensor networks. In *Proceedings of the 1st International Conference on Embedded Network Sensor Systems (SenSys '03)*, 63–75.

Liang, Q. and Ren, Q. 2005a. Energy and mobility aware geographical multipath routing for wireless sensor networks. In *2005 IEEE Wireless Communications and Networking Conference (WCNC'05)*. Vol. 3. IEEE, New Orleans, 1867–1871.

Liang, Q. and Ren, Q. 2005b. An energy-efficient MAC protocol for wireless sensor networks. In *2005 IEEE Global Telecommunications Conference (GLOBECOM'05)*. Vol. 1. IEEE, St. Louis.

Liu, J., Cheong, E., and Zhao, F. 2005. Semantics-based optimization across uncoordinated tasks in networked embedded systems. In *Proceedings of the 5th ACM International Conference On Embedded Software (EMSOFT 2005)*, W. Wolf, Ed. ACM, Jersey City, 273–281.

Liu, J., Zhang, Y., and Zhao, F. 2006. Robust distributed node localization with error management. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*. ACM, Florence, Italy, 250–261.

Liu, J. and Zhao, F. 2005. Towards semantic services for sensor-rich information systems. In *Proceedings of the 2nd International Conference on Broadband Networks (BROADNETS 2005)*. IEEE, Boston, 967–974.

Luo, H. and Pottie, G. 2005. A study on combined routing and source coding with explicit side information in sensor networks. In *2005 IEEE Global Telecommunications Conference (GLOBECOM'05)*. Vol. 5. IEEE, St. Louis.

Luo, R. C. and Kay, M. G. 1992. Data fusion and sensor integration: State-of-the-art 1990s. In *Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzalez, Eds. Academic Press, Inc., San Diego, CA, Chapter 3, 7–135.

Luo, R. C. and Kay, M. G., Eds. 1995. *Multisensor Integration and Fusion for Intelligent Machines and Systems*, Reissue edition Computer Engineering and Computer Science. Ablex Publishing, New Jersey, USA.

Luo, R. C., Yih, C.-C., and Su, K. L. 2002. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sensors J. 2,* 2 (April), 107–119.

Luo, X., Dong, M., and Huang, Y. 2006. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans. Comput. 55,* 1 (January), 2006.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. 2002. TAG: a Tiny AGgregation service for ad-hoc sensor networks. *ACM SIGOPS Oper. Syst. Rev. 36,* SI (Winter), 131–146.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. 2005. TinyDB: An acqusitional query processing system for sensor networks. *ACM Trans. Database Syst. 30,* 1 (March), 122–173.

Manzo, M., Roosta, T., and Sastry, S. 2005. Time synchronization attacks in sensor networks. In *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*. ACM, Alexandria, VA, 107–116.

Marco, D. and Neuhoff, D. L. 2004. Reliability vs. efficiency in distributed source coding for field-gathering sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 161–168.

Markin, M., Harris, C., Bernhardt, M., Austin, J., Bedworth, M., Greenway, P., Johnston, R., Little, A., and Lowe, D. 1997. Technology foresight on data fusion and data processing. Publication of The Royal Aeronautical Society.

MARZULLO, K. 1984. Maintaining the time in a distributed system: An example of a loosely-coupled distributed service. Ph.D. thesis, Stanford University, Department of Electrical Engineering, Stanford, CA.

MARZULLO, K. 1990. Tolerating failures of continuous-valued sensors. *ACM Trans. Comput. Syst. (TOCS) 8,* 4 (November), 284–304.

MASCOLO, C. AND MUSOLESI, M. 2006. SCAR: Context-aware adaptive routing in delay tolerant mobile sensor networks. In *Proceeding of the 2006 International Conference on Communications and Mobile Computing (IWCMC'06)*. ACM, Vancouver, Canada, 533–538.

MEGERIAN, S., KOUSHANFAR, F., QU, G., VELTRI, G., AND POTKONJAK, M. 2002. Exposure in wireless sensor networks: Theory and practical solutions. *Wireless Netw. 8,* 5 (September), 443–454.

MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M., AND SRIVASTAVA, M. 2001a. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of IEEE Infocom 2001*. Vol. 3. IEEE, Anchorage, AK, 1380–1387.

MEGUERDICHIAN, S., SLIJEPCEVIC, S., KARAYAN, V., AND POTKONJAK, M. 2001b. Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure. In *Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing*. ACM Press, Long Beach, CA, 106–116.

MEIER, L., BLUM, P., AND THIELE, L. 2004. Internal synchronization of drift-constraint clocks in ad-hoc sensor networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04)*. ACM, Tokyo, Japan, 90–97.

MHATRE, V. AND ROSENBERG, C. 2004. Homogeneous vs heterogeneous clustered sensor networks: A comparative study. In *Proceedings of the 2004 IEEE International Conference on Communications (ICC'04)*. Vol. 6. IEEE, Paris, France, 3646–3651.

MIGUEZ, J. AND ARTES-RODRIGUEZ, A. 2006. A Monte Carlo method for joint node location and maneuvering target tracking in a sensor network. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*. Vol. 4. IEEE, Toulouse, France.

MINI, R. A., LOUREIRO, A. A., AND NATH, B. 2004. The distinctive design characteristics of a wireless sensor network: The energy map. *Comput. Comm. 7,* 10 (June), 935–945.

MOONEY, R. J. 2000. Integrating abduction and induction in machine learning. In *Abduction and Induction, Essays on their Relation and Integration*, P. A. Flach and A. C. Kakas, Eds. Applied Logic Series. Kluwer, New York, Chapter 12, 336p.

MOSHIRI, B., ASHARIF, M. R., AND HOSEINNEZHAD, R. 2002. Pseudo information measure: A new concept for extension of Bayesian fusion in robotic map building. *Inform. Fusion 3,* 1 (March), 51–68.

NAKAMURA, E. F., DE OLIVEIRA, H. A., PONTELLO, L. F., AND LOUREIRO, A. A. 2006. On demand role assignment for event-detection in sensor networks. In *Proceedings of the 11th IEEE International Symposium on Computers and Communication (ISCC'06)*, P. Bellavista, C.-M. Chen, A. Corradi, and M. Daneshmand, Eds. IEEE Computer Society, Cagliari, Italy, 941–947.

NAKAMURA, E. F., FIGUEIREDO, C. M., AND LOUREIRO, A. A. 2005a. Information fusion for data dissemination in self-organizing wireless sensor networks. In *Proceedings of the 4th International Conference on Networking (ICN 2005)*, P. Lorenz and P. Dini, Eds. Lecture Notes in Computer Science, vol. 3420. Springer-Verlag GmbH, Reunion Island, France, 585–593.

NAKAMURA, E. F., NAKAMURA, F. G., FIGUEIREDO, C. M., AND LOUREIRO, A. A. 2005b. Using information fusion to assist data dissemination in wireless sensor networks. *Telecomm. Syst. 30,* 1–3 (November), 237–254.

NELSON, M. AND GAILLY, J.-L. 1995. *The Data Compression Book*, 2nd ed. M & T Books, New York, NY, USA.

NIU, R. AND VARSHNEY, P. K. 2006. Target location estimation in sensor networks with quantized data. *IEEE Trans. Sig. Proc. 54*, 12 (December), 4519–4528.

NORDLUND, P.-J., GUNNARSSON, F., AND GUSTAFSSON, F. 2002. Particle filters for positioning in wireless networks. In *Proceedings of the XI European Signal Processing Conference (EURSIPCO'02)*. Vol. II. TeSA, Toulouse, France, 311–314.

NOVÁK, V., PERFILIEVA, I., AND MOCKOR, J. 1999. *Mathematical Principles of Fuzzy Logic*. The International Series in Engineering and Computer Science. Kluwer Academic Publishers, Norwell, MA.

NOWAK, R., MITRA, U., AND WILLETT, R. 2004. Estimating inhomogeneous fields using wireless sensor networks. *IEEE J. Select. Areas Comm. 22,* 6 (August), 999–1006.

NOWAK, R. D. 2003. Distributed em algorithms for density estimation and clustering in sensor networks. *IEEE Trans. Sig. Proc. 51,* 8 (August), 2245–2253.

OLFATI-SABER, R. 2005. Distributed kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control(CDC'05)*. IEEE, Seville, Spain, 8179–8184.

PAGAC, D., NEBOT, E. M., AND DURRANT-WHYTE, H. 1998. An evidential approach to map-building for autonomous vehicles. *IEEE Trans. Robotics Autom. 14,* 4 (August), 623–629.

PAN, H., ANASTASIO, Z.-P., AND HUANG, T. 1998. A hybrid NN-Bayesian architecture for information fusion. In *Proceedings of the 1998 International Conference on Image Processing (ICIP'98)*. Vol. 1. IEEE, Chicago, IL, 368–371.

PATTEM, S., KRISHNAMACHARI, B., AND GOVINDAN, R. 2004. The impact of spatial correlation on routing with compression in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 28–35.

PATWARI, N., HERO, A. O., PERKINS, M., CORREAL, N. S., AND O'DEA, R. J. 2003. Relative location estimation in wireless sensor networks. *IEEE Trans. Sig. Proc. 51,* 8 (August), 2137–2148.

PEARL, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, USA.

PEIRCE, C. S. 1955. Abduction and induction. In *Philosophical Writings of Peirce*, C. S. Peirce and J. Buchler, Eds. Dover, New York, Chapter 11, 150–156.

PETROVIC, D., SHAH, R. C., RAMCHANDRAN, K., AND RABAEY, J. 2003. Data funneling: Routing with aggregation and compression for wireless sensor networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 2003)*. IEEE, Anchorage, AK, 156–162.

PINTO, A. J., STOCHERO, J. M., AND DE REZENDE, J. F. 2004. Aggregation-aware routing on wireless sensor networks. In *Proceedings of the IFIP TC6 9th International Conference on Personal Wireless Communications (PWC'04)*. Lecture Notes in Computer Science, vol. 3260. Springer, Delft, The Netherlands, 238–247.

POHL, C. AND VAN GENDEREN, J. L. 1998. Multisensor image fusion in remote sensing: Concepts, methods and applications. *Int. J. Remote Sens. 19,* 5 (March), 823–854.

POLASTRE, J., HILL, J., AND CULLER, D. 2004. Versatile low power media access for wireless sensor networks. In *SenSys'04*, J. A. Stankovic, A. Arora, and R. Govindan, Eds. ACM, Baltimore, MD, 95–107.

POOR, H. V. 1994. *An Introduction to Signal Detection and Estimation*, 2nd ed. Springer, New York.

POTTIE, G. J. AND KAISER, W. J. 2000. Wireless integrated network sensors. *Comm. ACM 43,* 5 (May), 51–58.

PRADHAN, S. S., KUSUMA, J., AND RAMCHANDRAN, K. 2002. Distributed compression in a dense microsensor network. *IEEE Sig. Proc. Mag. 19,* 2 (March), 51–60.

PRADHAN, S. S. AND RAMCHANDRAN, K. 2003. Distributed source coding using syndromes (DISCUS): design and construction. *IEEE Trans. Inform. Theory 49,* 3 (March), 626–643.

PROVAN, G. M. 1992. The validity of Dempster-Shafer belief functions. *Int. J. Approx. Reasoning 6,* 3 (May), 389–399.

PSOUNIS, K. 1999. Active networks: Applications, security, safety and architectures. *IEEE Comm. Surv. 2,* 1 (First Quarter), 2–16.

QI, H., WANG, X., IYENGAR, S. S., AND CHAKRABARTY, K. 2002. High performance sensor integration in distributed sensor networks using mobile agents. *Int. J. High Perf. Comput. Appl. 16,* 3 (Fall), 325–335.

RABBAT, M., HAUPT, J., SINGH, A., AND NOWAK, R. 2006. Decentralized compression and predistribution via randomized gossiping. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 51–59.

RABBAT, M. AND NOWAK, R. D. 2004. Distributed optimization in sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 20–27.

RACHLIN, Y., NEGI, R., AND KHOSLA, P. 2006. On the interdependence of sensing and estimation complexity in sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 160–167.

RAMAMOORTHY, A., JAIN, K., CHOU, P. A., AND EFFROS, M. 2006. Separating distributed source coding from network coding. *IEEE Trans. Inf. Theory 52,* 6 (June), 2785–2795.

RAMCHANDRAN, K., SZTIPANOVITS, J., HOU, J. C., AND PAPPAS, T. N., Eds. 2004. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*. ACM, Berkeley, CA.

RANGWALA, S., GUMMADI, R., GOVINDAN, R., AND PSOUNIS, K. 2006. Interference-aware fair rate control in wireless sensor networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'06)*. ACM, Pisa, Italy, 63–74.

RAO, N. S. V. 2001. On fusers that perform better than the best sensor. *IEEE Trans. Patt. Anal. Mach. Intell. 23,* 8 (August), 904–909.

RATNASAMY, S., KARP, B., SHENKER, S., ESTRIN, D., GOVINDAN, R., YIN, L., AND YU, F. 2003. Data-centric storage in sensornets with GHT, a geographic hash table. *Mobile Networks and Applications (MONET) 8,* 4 (August), 427–442.

RAVIRAJ, P., SHARIF, H., HEMPEL, M., AND CI, S. 2005. MOBMAC—an energy efficient and low latency MAC for mobile wireless sensor networks. In *Proceedings of the 2005 Systems Communications*. IEEE, Montreal, Canada, 370–375.

REDI, J., BALAKRISHNAN, H., AND ZHAO, F., Eds. 2005. *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*. ACM, San Diego, CA.

RHEE, I., WARRIER, A., AIA, M., AND MIN, J. 2005. Z-mac: A hybrid MAC for wireless sensor networks. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*, 90–101.

RIBO, M. AND PINZ, A. 2001. A comparison of three uncertainty calculi for building sonar-based occupancy grids. *Robotics and Autonomous Systems 35,* 3–4 (June), 201–209.

RÖMER, K., BLUM, P., AND MEIER, L. 2005. Time synchronization and calibration in wireless sensor networks. In *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, Ed. John Wiley & Sons, Hoboken, NJ, 199–237.

ROSENBLATT, F. 1959. Two theorems of statistical separability in the perceptron. In *Mechanization of Thought Processes*. National Physical Laboratory, London, UK, 421–456.

ROTH, M. R. 1990. Survey of neural network technology for automatic target recognition. *IEEE Trans. Neural Netw. 1,* 1 (March), 28–33.

SALIGRAMA, V., ALANYALI, M., AND SAVAS, O. 2006. Distributed detection in sensor networks with packet losses and finite capacity links. *IEEE Trans. Sig. Proc. 54,* 11 (November), 4118–4132.

SAM, D., NWANKPA, C., AND NIEBUR, D. 2001. Decision fusion of voltage stability indicators for small sized power systems. In *IEEE Power Engineering Society Summer Meeting*. Vol. 3. IEEE, Vancouver, British Columbia, Canada, 1658–1663.

SANTINI, S. AND RÖMER, K. 2006. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006)*. TRF, Chicago, IL, 29–36.

SAVVIDES, A., HAN, C., AND STRIVASTAVA, M. B. 2003. The n-hop multilateration primitive for node localization. *Mobile Netw. Appl. 8,* 4 (August), 443–451.

SCAGLIONE, A. AND SERVETTO, S. D. 2002. On the interdependence of routing and data compression in multihop sensor networks. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*. ACM, Atlanta, GA, 140–147.

SCHMID, U. AND SCHOSSMAIER, K. 2001. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. *Distrib. Comput. 14,* 2 (April), 101–111.

SCHMITT, T., HANEK, R., BEETZ, M., BUCK, S., AND RADIG, B. 2002. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Trans. Robotics Autom. 18,* 5 (October), 670–684.

SHAFER, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ.

SHAH, S. F. A., RIBEIRO, A., AND GIANNAKIS, G. B. 2005. Bandwidth-constrained MAP estimation for wireless sensor networks. In *Conference Record of the 39th Asilomar Conference on Signals, Systems and Computers*. IEEE, Pacific Grove, CA, 215–219.

SHARAF, M. A., BEAVER, J., LABRINIDIS, A., AND CHRYSANTHIS, P. K. 2003. TiNA: A scheme for temporal coherency-aware in-network aggregation. In *Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM Press, San Diego, CA, 69–76.

SHENG, B., LI, Q., AND MAO, W. 2006. Data storage placement in sensor networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'06)*. ACM, Florence, Italy, 344–355.

SHENG, X. AND HU, Y.-H. 2005. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Trans. Sig. Proc. 53,* 1 (January), 44–53.

SHENG, X., HU, Y. H., AND RAMANATHAN, P. 2005. Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, 181–188.

SHU, H. AND LIANG, Q. 2005. Fuzzy optimization for distributed sensor deployment. In *2005 IEEE Wireless Communications and Networking Conference (WCNC'05)*. Vol. 3. IEEE, New Orleans, LA, 1903–1908.

SHULSKY, A. N. AND SCHMITT, G. J. 2002. *Silent Warfare: Understanding the World of Intelligence*, 3 ed. Brasseys, Inc., New York, NY.

SIATERLIS, C. AND MAGLARIS, B. 2004. Towards multisensor data fusion for DoS detection. In *Proceedings of the 2004 ACM Symposium on Applied Computing*. ACM Press, Nicosia, Cyprus, 439–446.

SICHITIU, M. L. AND RAMADURAI, V. 2004. Localization of wireless sensor networks with a mobile beacon. In *Proceedings of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2004)*. IEEE, Fort Lauderdale, FL, 174–183.

SIDHU, T., CRUDER, O., AND HUFF, G. 1997. An abductive inference technique for fault diagnosis in electrical power transmission networks. *IEEE Trans. Power Deliv. 12,* 1 (January), 515–522.

SINGH, A., NOWAK, R., AND RAMANATHAN, P. 2006. Active learning for adaptive mobile sensing networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 60–68.

SINOPOLI, B., SCHENATO, L., FRANCESCHETTI, M., POOLLA, K., JORDAN, M., AND SASTRY, S. 2004. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Cont. 49,* 9 (September), 1453–1464.

SMITH, S. W. 1999. *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing, San Diego, CA.

SOHRABI, K., GAO, J., AILAWADHI, V., AND POTTIE, G. J. 2000. Protocols for self-organization of a wireless sensor network. *IEEE Pers. Comm. 7,* 5 (October), 16–27.

SPANOS, D., OLFATI-SABER, R., AND MURRAY, R. M. 2005. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, 133–139.

SRINIVASAN, T., CHANDRASEKAR, R., AND VIJAYKUMAR, V. 2006. A fuzzy, energy-efficient scheme for data centric multipath routing in wireless sensor networks. In *2006 IFIP International Conference on Wireless and Optical Communications Networks*. IEEE, Bangalore, India.

STANKOVIC, J. A., GIBBONS, P. B., WICKER, S. B., AND PARADISO, J. A., Eds. 2006. *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*. ACM, Nashville, TN.

STANN, F. AND HEIDEMANN, J. 2005. BARD: Bayesian-assisted resource discovery in sensor networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 866–877.

STEINBERG, A. N., BOWMAN, C. L., AND WHITE, F. E. 1999. Revisions to the JDL data fusion model. In *Proceedings of the SPIE*. Vol. 3719. SPIE, Orlando, FL, 430–441.

TANG, C. AND RAGHAVENDRA, C. S. 2005. Wavelet based source broadcast for in-network processing in sensor networks unknown side information. In *2005 IEEE Global Telecommunications Conference (GLOBECOM'05)*. Vol. 1. IEEE, St. Louis, USA.

TANG, C., RAGHAVENDRA, C. S., AND PRASANNA, V. K. 2003. An energy efficient adaptive distributed source coding scheme in wireless sensor networks. In *Proceedings of the 2003 IEEE International Conference on Communications (ICC'03)*. Vol. 1. IEEE, Anchorage, AK, 732–737.

TENNEY, R. R. AND SANDELL JR., N. R. 1981. Detection with distributed sensors. *IEEE Trans. Aerosp. Electron. Syst. 17,* 4 (July), 501–510.

TIAN, D. AND GEORGANAS, N. D. 2002. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*. ACM Press, Atlanta, GA, 32–41.

TSYMBAL, A., PUURONEN, S., AND PATTERSON, D. W. 2003. Ensemble feature selection with the simple Bayesian classification. *Inform. Fusion 4,* 2 (June), 87–100.

U.S. DEPARTMENT OF DEFENSE. 1991. Data fusion lexicon. Published by Data Fusion Subpanel of the Joint Directors of Laboratories. Tecnichal Panel for C3 (F.E. White, Code 4202, NOSC, San Diego, CA).

VAN RENESSE, R. 2003. The importance of aggregation. In *Future Directions in Distributed Computing: Research and Position Papers*, A. Schiper, A. A. Shvartsman, H. Weatherspoon, and B. Y. Zhao, Eds. Lecture Notes in Computer Science, vol. 2584. Springer, Bologna, Italy, 87–92.

VARSHNEY, P. K. 1997. *Distributed Detection and Data Fusion*. Springer, New York, USA.

VERCAUTEREN, T., GUO, D., AND WANG, X. 2005. Joint multiple target tracking and classification in collaborative sensor networks. *IEEE J. Sel. Areas Comm. 23,* 4 (April), 714–723.

WAGNER, R. S., BARANIUK, R. G., DU, S., JOHNSON, D. B., AND COHEN, A. 2006. An architecture for distributed wavelet analysis and processing in sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 243–250.

WALD, L. 1999. Some terms of reference in data fusion. *IEEE Trans. Geosci. Remote Sens. 13,* 3 (May), 1190–1193.

WALLACE, J., PESCH, D., REA, S., AND IRVINE, J. 2005. Fuzzy logic optimisation of MAC parameters and sleeping duty-cycles in wireless sensor networks. In *62nd Vehicular Technology Conference, 2005. VTC-2005-Fall*. Vol. 3. IEEE, Dallas, TX, 1824–1828.

WALTZ, E. L. AND LLINAS, J. 1990. *Multisensor Data Fusion*. Artech House, Norwood, MA.

WELCH, G. AND BISHOP, G. 2001. An introduction to the Kalman filter. In *SIGGRAPH 2001 Course Notes*. ACM, Los Angeles, CA. Course 8.

WHITEHOUSE, K., LIU, J., AND ZHAO, F. 2006. Semantic streams: A framework for composable inference over sensor data. In *3rd European Workshop on Wireless Sensor Networks (EWSN'06)*, K. Römer, H. Karl, and F. Mattern, Eds. Lecture Notes in Computer Science, vol. 3868. Springer, Zurich, Switzerland, 5–20.

WIDROW, B. AND HOFF, M. E. 1960. Adaptive switching circuits. *1960 IRE Western Electric Show and Convention Record 4*, 96–104.

WILLETT, R., MARTIN, A., AND NOWAK, R. 2004. Backcasting: Adaptive sampling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 124–133.

WONG, Y., WU, J., NGOH, L., AND WONG, W. 2004. Collaborative data fusion tracking in sensor networks using monte carlo methods. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*. IEEE, Tampa, FL, 563–564.

WONGNGAMNIT, C. AND ANGLUIN, D. 2001. Robot localization in a grid. *Inform. Proc. Lett. 77,* 5–6 (March), 261–267.

WOO, A., TONG, T., AND CULLER, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Network Sensor Systems (SenSys'03)*, 14–27.

WU, Q., RAO, N. S., BARHEN, J., IYENGAR, S. S., VAISHNAVI, V. K., QI, H., AND CHAKRABARTY, K. 2004. On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Trans. Knowl. Data Eng. 16,* 6 (June), 740–753.

XIAO, J., RIBEIRO, A., LUO, Z., AND GIANNAKIS, G. B. 2006a. Distributed compression-estimation using wireless sensor networks. *IEEE Sig. Proc. Mag. 23,* 4 (July), 27–41.

XIAO, L., BOYD, S., AND LALL, S. 2005. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, 63–70.

XIAO, L., BOYD, S., AND LALL, S. 2006b. A space-time diffusion scheme for peer-to-peer least-squares estimation. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, 168–176.

XIONG, Z., LIVERIS, A. D., AND CHENG, S. 2004. Distributed source coding for sensor networks. *IEEE Sig. Proc. Mag. 21,* 5 (September), 80–94.

XU, Y. AND QI, H. 2004. Distributed computing paradigms for collaborative signal and information processing in sensor networks. *J. Para. Distrib. Comput. 64,* 8 (August), 945–959.

YANG, C.-L., BAGCHI, S., AND CHAPPELL, W. J. 2005a. Location tracking with directional antennas in wireless sensor networks. In *2005 IEEE MTT-S International Microwave Symposium Digest*. IEEE, Long Beach, CA.

YANG, Z., GUO, D., AND WANG, X. 2005b. Blind decoding of multiple description codes over OFDM systems via sequential Monte Carlo. *EURASIP J. Wireless Comm. Netw. 2005,* 2, 141–154.

YAO, Y. AND GEHRKE, J. 2002. The cougar approach to in-network query processing in sensor networks. *Sigmod Rec. 31,* 3 (September), 9–18.

YIYAO, L., VENKATESH, Y. V., AND KO, C. C. 2001. A knowledge-based neural network for fusing edge maps of multi-sensor images. *Inform. Fusion 2,* 2 (June), 121–133.

YU, B., SYCARA, K., GIAMPAPA, J. A., AND OWENS, S. R. 2004. Uncertain information fusion for force aggregation and classification in airborne sensor networks. In *AAAI-04 Workshop on Sensor Networks*. AAAI Press, San Jose, CA.

YUAN, Y. AND KAM, M. 2004. Distributed decision fusion with a random-access channel for sensor network applications. *IEEE Trans. Instr. Meas. 53,* 4 (August), 1339–1344.

YUEN, D. C. K. AND MACDONALD, B. A. 2002. A comparison between extended Kalman filtering and sequential Monte Carlo techniques for simultaneous localisation and map-building. In *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, W. Friedrich and P. Lim, Eds. ARAA, Auckland, New Zealand, 111–116.

YUSUF, M. AND HAIDER, T. 2005. Energy-aware fuzzy routing for wireless sensor networks. In *IEEE International Conference on Emerging Technologies (ICET'05)*. IEEE, Islamiabad, Pakistan, 63–69.

ZADEH, L. A. 1994. Fuzzy logic and soft computing: Issues, contentions and perspectives. In *Proceedings of the 3rd International Conference on Fuzzy Logic, Neural Nets and Soft Computing*. Fuzzy Logic Systems Institute, Iisuka, Japan, 1–2.

ZENG, Z. AND MA, S. 2002. Head tracking by active particle filtering. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*. IEEE, Washington, D.C., USA, 82–87.

ZHANG, X. AND WICKER, S. B. 2005. Robustness vs. efficiency in sensor networks. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05)*, 225–230.

ZHAO, F. AND GUIBAS, L. J., Eds. 2003. *Information Processing in Sensor Networks: 2nd International Workshop (IPSN'03)*. Lecture Notes in Computer Science, vol. 2634. Springer, Palo Alto, CA.

Zhao, F., Liu, J., Liu, J., Guibas, L., and Reich, J. 2003a. Collaborative signal and information processing: An information directed approach. *Proc. IEEE 91,* 8 (August), 1199–1209.

Zhao, F., Shin, J., and Reich, J. 2002a. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Sig. Proc. Mag. 19,* 2 (March), 61–72.

Zhao, J., Govindan, R., and Estrin, D. 2002b. Residual energy scans for monitoring wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'02)*. Vol. 1. IEEE, Orlando, FL, 356–362.

Zhao, J., Govindan, R., and Estrin, D. 2003b. Computing aggregates for monitoring wireless sensor networks. In *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 2003)*. IEEE, Anchorage, AK, 139–148.

Zhou, B., Ngoh, L. H., Lee, B. S., and Fu, C. P. 2004. A hierarchical scheme for data aggregation in sensor networks. In *Proceedings of the 12th IEEE International Conference on Networks (ICON'04)*. Vol. 2. IEEE, Singapore, 525–529.

Zhou, C. and Krishnamachari, B. 2003. Localized topology generation mechanisms for self-configuring sensor networks. In *2003 IEEE Global Telecommunications Conference (GLOBECOM'03)*. Vol. 22. IEEE, San Francisco, CA, 1269–1273.

Zhu, Y., Vedantham, R., Park, S.-J., and Sivakumar, R. 2005. A scalable correlation aware aggregation strategy for wireless sensor networks. In *Proceedings of the 1st International Conference on Wireless Internet (WICON'05)*. IEEE, Budapest, Hungary, 122–129.

Received December 2005; revised December 2006; accepted March 2007