

Shubha Kadambe<sup>1</sup>

<sup>1</sup>HRL Laboratories, LLC, 3011 Malibu Canyon Rd., Malibu, CA 90265, USA

# Power and Resource Aware Distributed Smart Fusion

Published in:

Florentin Smarandache & Jean Dezert (Editors)

**Advances and Applications of DSMT for Information Fusion**

(Collected works), Vol. I

American Research Press, Rehoboth, 2004

ISBN: 1-931233-82-9

Chapter XVIII, pp. 383 - 409

**Abstract:** *Large distributed sensor networks (DSN) with disparate sensors, processors and wireless communication capabilities are being developed for a variety of commercial and military applications. Minimizing power consumption of the nodes is a critical issue to their good functioning during the mission or application, to reduce their size and weight, and their cost so that their deployment is economically viable. In this chapter, we describe a robust, flexible, and distributed smart fusion algorithm that provides high decision accuracy and minimizes power consumption through efficient use of network sensing, communication, and processing resources. Our approach, developed on information theory-based metrics, determines what network resources (sensors, platforms, processing, and communication) are necessary to accomplish mission tasks, then uses only those necessary resources. It minimizes the network power consumption and combines valuable information at features and decision level using DSMT. We demonstrate the proposed optimal, fully autonomous, smart distributed fusion algorithm for target detection and classification using a DSN. Our experimental results show that our approach significantly improves the detection and classification accuracy using the required high quality sensors and features, and valuable fused information.*

The part of this work was funded by DARPA/IXO under the contract # F30602-01-C-0192. The authors would like to thank in particular Dr. Sri Kumar for funding this work under the SensIT program.

## 18.1 Introduction

**S**patially distributed network of inexpensive, small and smart nodes with multiple onboard sensors is an important class of emerging networked systems for various defense and commercial applications. Since this network of sensors has to operate efficiently in adverse environments using limited battery power and resources, it is important that appropriate sensors process information hierarchically and share information only if it is valuable in terms of improving the decision accuracy such that highly accurate decision is made progressively. One way to address this problem is to activate only those sensors that provide missing and relevant information, to assess the quality of information obtained from the activated sensors (this helps in determining the sensor quality), to assess the value of obtained information in terms of improving the decision (e.g., target detection/track) accuracy, to communicate only relevant, high quality and valuable information to the neighboring nodes and to fuse only valuable information that aid in progressive decisions. Information theoretic approaches provide measures for relevance, utility, missing information, value of information, etc. These measures help in achieving hierarchical extraction of relevant and high quality of information that enable in selection/actuation of relevant sensors and dynamically discard information from noisy or dead sensors and, progressive improvement of decision accuracy and confidence by utilizing only valuable information while fusing information obtained from neighboring nodes. In this chapter, we describe a minmax entropy based technique for missing information (feature) and information type (sensor) discovery, within class entropy based technique for sensor discrimination (i.e., quality assessment), mutual information for features quality assessment and, mutual information and other measures for assessing the value of information in terms of improvement in decision accuracy. In addition, we briefly describe how high quality, relevant and valuable information is fused using a new theory – DSMT which provides rules for combining two or more masses of independent sources of information that is dynamically changing in real time which is essential in the network of disparate sensors that is considered here.

To the best knowledge of this author there is no study on sensor discrimination using within class entropy metric is reported even though, there is one study on using mutual information for selecting a subset of features from a bigger set that is described in [2]. The technique described in this chapter uses within class entropy as a metric to assess the quality (good vs. bad) of a sensor. Unlike our technique, the technique in [2] is static in nature and cannot handle the case where the dimensionality of the feature set varies. In [15], the author shows that in general by fusing data from selective sensors the performance of a network of sensors can be improved. However, in this study, no specific novel metrics for the feature discovery and feature/sensor discrimination were developed unlike in this chapter. In [10], techniques to represent Kalman filter state estimates in the form of information – Fisher and Shannon entropy are provided. In such a representation it is straightforward to separate out what is new information from what is

either prior knowledge or common information. This separation procedure is used in decentralized data fusion algorithms that are described in [10]. However, to the best knowledge of this author no study has been reported on using minmax entropy principle for the feature and information type discovery. Furthermore, to our knowledge the proposed value of information based fusion is not studied by others and is another significant contribution of this chapter. In addition, the significance of this study is the application of feature discovery and sensor discrimination in awakening the required sensor and in the formation of a cluster of distributed sensors to reduce the power consumption, to improve the decision accuracy and to reduce the communication bandwidth requirements. This chapter is a comprehensive of our studies reported in [6, 7, 8] with the addition of application of DSmT for fusion at both feature and decision levels.

In the next section, proposed techniques are described. The simulation description and experimental results are provided in section 18.3. Conclusions and future research directions are provided in section 18.4.

## 18.2 Description of proposed research

### 18.2.1 Discovery of missing information

In the case of applications of a distributed network of disparate sensors such as (a) target detection, identification and tracking, (b) classification, (c) coalition formation, etc., the missing information could correspond to feature discovery. This helps in only probing (awakening) the sensor node that can provide the missing information and thus save power and processing by not arbitrarily activating nodes and by letting the unused sensor be in the sleep mode. We apply the minmax entropy principle described in [9] for the feature discovery. The details of estimation of missing information in other words feature discovery and information type using the minmax entropy principle are as follows.

#### 18.2.1.1 Minmax entropy principle

Let  $N$  given values corresponds to  $n$  different information types. Let  $z_{ij}$  be the  $j$ -th member of  $i$ -th information type (where the information type is defined as a sensor type that gives similar information measures) so that

$$i = 1, 2, \dots, n \quad j = 1, 2, \dots, m_i \quad \sum_{i=1}^n m_i = N \quad (18.1)$$

Then the entropy for this type of classes of information is:

$$H = - \sum_{i=1}^n \sum_{j=1}^{m_i} \frac{z_{ij}}{T} \ln \frac{z_{ij}}{T} \quad \text{where} \quad T = \sum_{i=1}^n \sum_{j=1}^{m_i} z_{ij} \quad (18.2)$$

Let  $T_i = \sum_{j=1}^{m_i} z_{ij}$ . Using this,  $H$  can be written as:

$$H = \sum_{i=1}^n \frac{T_i}{T} H_i - \sum_{i=1}^n \frac{T_i}{T} \ln \frac{T_i}{T} = H_W + H_B \quad (18.3)$$

where  $H_i = - \sum_{j=1}^{m_i} \frac{z_{ij}}{T} \ln \frac{z_{ij}}{T}$  is the entropy of values that belong to information  $i$ .

In the equation above,  $H_W$  and  $H_B$  are entropy of *within classes* (information types) and *between classes*, respectively. We would like types of information to be as distinguishable as possible and we would like the information within each type to be as homogenous as possible. The entropy is high if the values belonging to a type (class) represent similar information and is low if they represent dissimilar information. Therefore, we would like  $H_B$  to be as small as possible and  $H_W$  as large as possible. This is the principle of minmax entropy.

### 18.2.1.2 Application of minimax entropy principle for feature discovery

Let  $z$  be the missing value (feature). Let  $T$  be the total of all known values such that the total of all values is  $T + z$ . Let  $T_1$  be the total of values that belong to information type to which  $z$  may belong.  $T_1 + z$  then is the total of that particular type of information. This leads to:

$$\begin{cases} H = - \sum' \frac{z_{ij}}{T+z} \ln \frac{z_{ij}}{T+z} - \frac{z}{T+z} \ln \frac{z}{T+z} \\ H_B = - \sum'' \frac{T_i}{T+z} \ln \frac{T_i}{T+z} - \frac{T_1+z}{T+z} \ln \frac{T_1+z}{T+z} \end{cases} \quad (18.4)$$

Here  $\sum'$  denotes the summation over all values of  $i, j$  except that correspond to the missing information and  $\sum''$  denotes over all values of  $i$  except for the type to which the missing information belongs, respectively.

We can then estimate  $z$  by minimizing  $H_B/H_W$  or  $H_B/(H - H_B)$  or  $H_B/H$ , or by maximizing  $(H - H_B)/H_B$  or  $H/H_B$ . The estimates of  $z$  provide the missing information values (features) and information (sensor) type. From the above discussion, we can see that we will be able to discover features as well as type of sensor from which these features can be obtained. This has the advantage of probing the appropriate sensor in a DSN. The transfer of information and probing can be achieved in such a network by using network routing techniques. Before trying to use the newly acquired feature set from the estimated information type i.e., sensor, it is advisable to check the quality of the sensor to make sure that the sensor from which we are seeking the information is not noisy (not functioning properly) or “dead” to reduce the cost of processing. In a DSN this has an added advantage of reducing the communication cost. We measure (see next section) the quality (i.e. discriminate a good sensor vs. bad sensor) by using an information theoretic measure - the within class entropy.

### 18.2.2 Measure of consistency

We measure relevance by measuring consistency. For this we have developed a metric based on within class entropy that is described in this section. Let there are  $N$  events (values) that can be classified in to  $m$  classes and let an event  $x_{ij}$  be the  $j$ -th member of  $i$ -th where  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n_i$  and  $\sum_{i=1}^m n_i = N$ . The entropy for this classification is:

$$\begin{aligned}
 H &= \sum_{i=1}^m \sum_{j=1}^{n_i} p(i)p(x_{ij}) \log\left(\frac{1}{p(i)p(x_{ij})}\right) \\
 &= - \sum_{i=1}^m \sum_{j=1}^{n_i} p(i)p(x_{ij}) \log(p(i)p(x_{ij})) \\
 &= - \sum_{i=1}^m p(i) \sum_{j=1}^{n_i} p(x_{ij}) \log(p(x_{ij})) - \sum_{i=1}^m p(i) \log(p(i)) \sum_{j=1}^{n_i} p(x_{ij}) \\
 &= \sum_{i=1}^m p(i)H_i - \sum_{i=1}^m p(i) \log(p(i)) \\
 &= H_W + H_B
 \end{aligned}$$

The penultimate equality comes from the definition of  $H_i = - \sum_{j=1}^{n_i} p(x_{ij}) \log(p(x_{ij}))$  representing the entropy of a class  $i$  and the total probability theorem, i.e.  $\sum_{j=1}^{n_i} p(x_{ij}) = 1$ .  $H_W$  is called the entropy *within* classes and  $H_B$  is called the entropy *between* classes.

The entropy  $H_W$  is high if the values or events belonging to a class represent similar information and is low if they represent dissimilar information. This means  $H_W$  can be used as a measure to define consistency. That is, if two or more sensor measurements are similar then their  $H_W$  is greater than if they are dissimilar. Therefore, this measure can be used in sensor discrimination. Note that even though the definitions of within class and between class entropy here are slightly different from section 18.2.1, they are similar in concept. Note also that the minmax entropy measure that uses both within and between class entropies was used earlier in the estimation of missing information; but here, within class entropy is defined as a consistency measure that can be used in sensor discrimination or selection. These two metrics have different physical interpretations and are used for different purposes.

### 18.2.3 Feature discrimination

After making sure about the quality of sensor (the information type) from which missing information can be obtained, it is necessary to make sure that the observations (features) from that sensor does help in gaining information as far as the required decision is concerned. This step doubly makes sure that the estimated missing information is indeed needed. For this, we have developed metrics based on conditional entropy and mutual information which are described in the following two subsections.

### 18.2.3.1 Conditional entropy and mutual information

Entropy is a measure of uncertainty. Let  $H(x)$  be the entropy of previously observed  $x$  events. Let  $y$  be a new event. We can measure the uncertainty of  $x$  after including  $y$  by using the conditional entropy which is defined as:

$$H(x|y) = H(x, y) - H(y) \quad (18.5)$$

with the property  $0 \leq H(x|y) \leq H(x)$ . The conditional entropy  $H(x|y)$  represents the amount of uncertainty remaining about  $x$  after  $y$  has been observed. If the uncertainty is reduced then there is information gained by observing  $y$ . Therefore, we can measure the importance of observing estimated  $y$  by using conditional entropy. Another measure that is related to conditional entropy that one can use is the mutual information  $I(x, y)$  which is a measure of uncertainty that is resolved by observing  $y$  and is defined as:

$$I(x, y) = H(x) - H(x|y) \quad (18.6)$$

To explain how this measure can be used to measure the importance of estimated missing information (e.g., features) which is referred to as feature discrimination, an example is provided below.

### 18.2.3.2 Example of feature discrimination based on entropy metrics

Let  $A = \{a_k\}$ ,  $k = 1, 2, \dots$  be the set of features from sensor 1 and let  $B = \{b_l\}$ ,  $l = 1, 2, \dots$  be the set of features from sensor 2. Let  $p(a_k)$  be the probability of feature  $a_k$  and  $p(b_l)$  the probability of feature  $b_l$ . Let  $H(A)$ ,  $H(B)$  and  $H(A|B)$  be the entropy corresponding to sensor 1, sensor 2 and sensor 1 given sensor 2, respectively, and they are defined as [9]:

$$\begin{cases} H(A) = \sum_k p(a_k) \log\left(\frac{1}{p(a_k)}\right) \\ H(A|B) = H(A, B) - H(B) = \sum_l p(b_l) H(A|b_l) = \sum_l p(b_l) \sum_k p(a_k|b_l) \log\left(\frac{1}{p(a_k|b_l)}\right) \end{cases} \quad (18.7)$$

Here, the entropy  $H(A)$  corresponds to the prior uncertainty and the conditional entropy  $H(A|B)$  corresponds to the amount of uncertainty remaining after observing features from sensor 2. The mutual information  $I(A, B) = H(A) - H(A|B)$  corresponds to uncertainty that is resolved by observing  $B$  in other words features from sensor 2. From the definition of mutual information, it can be seen that the uncertainty that is resolved basically depends on the conditional entropy. Let us consider two types of sensors at node 2. Let the set of features of these two sensors be  $B_1$  and  $B_2$ , respectively and let the set of features estimated by the minmax entropy principle described in the previous section be  $B_1$ . If  $H(A|B_1) < H(A|B_2)$  then  $I(A, B_1) > I(A, B_2)$ . This implies that the uncertainty is better resolved by observing  $B_1$  as compared to  $B_2$ . This further implies that indeed the estimated  $B_1$  corresponds to features that help in gaining information that aid in the decision process of sensor 1 and  $B_2$  does not and hence, should not be considered.

Note that even though in the above example only two sensor nodes are considered for simplicity, this measure or metric can be used in a network of more than two sensors. In such a case,  $A$  would be a set of features that a node already has from other sensors in a cluster that it is a member of and  $B$  would be a new feature set that it receives from a different sensor type that it has not already received from and it may be a member or not a member of that particular cluster. If the mutual information increases by including the set of features  $B$  then we make a decision of including that sensor as part of this particular cluster if it is not a member. In case it is a member and the mutual information does not increase then it would be discarded from that particular cluster.

### 18.2.4 Measures of value of information

This section describes the measures of value of information that we have developed to determine when to fuse information from disparate sources. The value is in terms of improving the decision accuracy. Even though the mathematics of the metrics described below are not novel, the usage of metrics in the context of verifying value of information with respect to improving the decision accuracy (e.g., classification accuracy, detection accuracy) is new.

#### 18.2.4.1 Mutual information

Mutual information defined in section 18.2.3.1 can also be used as a measure of value.

#### 18.2.4.2 Euclidean Distance

Unlike mutual information, Euclidean distance does not evaluate the amount of information available from a second source. It does, however, measure the similarity between two feature sets in Euclidean space. This value can then be used to determine when to fuse two sources of information, whether they are from different types of sensors on the same node or from same type of sensors different nodes. A simple measure, Euclidean distance is defined as:

$$d = \sqrt{\sum_i (a_i - b_i)^2} \quad (18.8)$$

where  $a_i$ ,  $b_i$  and  $i$  are defined in Section 18.2.3.1.

#### 18.2.4.3 Correlation

Correlation is also a well known measure of similarity. We use the standard measure of correlation as defined by:

$$\rho = \frac{E[(a - \mu_a)(b - \mu_b)]}{E[a - \mu_a]E[b - \mu_b]} \quad (18.9)$$

where  $\mu_a$  and  $\mu_b$  are the means of feature sets  $a$  and  $b$ , respectively. Note that correlation is very closely related to mutual information,  $I(x, y)$  because (18.6) can be rewritten as:

$$I(x, y) = \sum_k p(a_k, b_k) \log\left(\frac{p(a_k, b_k)}{p(a_k)p(b_k)}\right) \quad (18.10)$$

#### 18.2.4.4 Kullback-Liebler distance

Finally, the Kullback-Liebler (KL) distance is derived from entropy, and again is a measure of the separation of two feature sets. It is defined as:

$$D = \sum_k p(a_k) \log\left(\frac{p(a_k)}{p(b_k)}\right) + \sum_k p(b_k) \log\left(\frac{p(b_k)}{p(a_k)}\right) \quad (18.11)$$

### 18.2.5 Fusion using DSMT

Since in a network of disparate sensor nodes as is considered here, the sources of information are independent and changing dynamically based on which sensor and features are selected, for the smart distributed fusion we use the new theory of plausible and paradoxical reasoning – DSMT developed in [5]. This theory provides a hybrid DSMT rule which combines or fuses two or more masses of independent sources of information and takes care of restraints i.e., of sets which might become empty at certain time or new sets that might arise at some other time. In a network of sensor nodes these situations arise (sometimes we discard the feature set or decision from the other nodes and sometimes we use features from different type of sensors based on how the scene is changing dynamically) and hence, the application of hybrid DSMT rule for fusion is very appropriate. In addition, since fusion is not done at a centralized location but done locally dynamically based on the information received from the neighboring nodes, we propose to extend the decentralized dynamical fusion by combining dynamical fusion using the hybrid DSMT rule for the chosen hybrid model  $\mathcal{M}$ . Specifically, at the feature level fusion at each sensor node the frame under consideration at time  $t_l$  will be

$$\Theta(t_l) \triangleq \{\theta_1 = \text{acoustic sensor}, \theta_2 = \text{seismic sensor}, \theta_3 = \text{IR sensor location}\}$$

and at decision level fusion,  $\Theta(t_l) = \{\theta_1 = \text{vehicle present}, \theta_2 = \text{vehicle not present}\}$  in the case of a detection application and,

$$\Theta(t_l) \triangleq \{\theta_1 = \text{AAV}, \theta_2 = \text{DW}, \theta_3 = \text{HMMWV}\}$$

where AAV, DW, and HMMWV represent the vehicle types that are being classified) for the decision level fusion in the case of a classification application.

Both detection and classification applications are described in section 18.3.2. We derive basic belief assignments based on the observations (a) from the sensor type for feature level fusion, (b) from the

features extracted from the sensors' signals for fusion at the decision level in the case of classification and detection applications. For example,  $m_a(\theta_1) = 0$  and  $m_a(\theta_2) = 0$ , if the feature from the acoustic sensor (a) – energy is well above the threshold level in the case of the detection application.  $\Theta(t_l)$  changes as the observation is different based on the above described sensor and feature selection and results in  $\Theta(t_{l+1})$ . If we discard observations from a sensor (based on the feature discrimination algorithm explained above) then  $\Theta$  diminishes and we apply the hybrid DSm rule to transfer the masses of empty sets to non-empty sets. If we include observations from a new sensor then we use the classical DSm fusion rule to generate basic belief assignments  $m_{t_{l+1}}(.)$ . For the decentralized decision level fusion at the current node, consider the  $\Theta_p(t_l)$  obtained from the previous node and the  $\Theta(t_l)$  of the current node and apply the hybrid DSm rule by taking the integrity constraints in to consideration. These constraints are generated differently for the fusion between the sensors and for the fusion from node to node. The pseudo-codes which generate these constraints are given in section 18.3.2. For example, in the case of node to node fusion for classification application that is described in section 18.3.2.2.1 `fuse_4class=1` will indicate to put the constraint  $\theta_1 \cap \theta_2 \stackrel{M}{=} \emptyset$ ,  $\theta_1 \cap \theta_3 \stackrel{M}{=} \emptyset$ ,  $\theta_1 \cap \theta_2 \cap \theta_3 \stackrel{M}{=} \emptyset$  at the current node if the classification at the previous node corresponds to  $\theta_1 = \text{AAV}$  since if the vehicle at the previous node is AAV, the vehicle at the current node which is very close to the previous node has to be AAV.

## 18.3 Experimental details and results

Above described algorithms have been applied for the feature discovery, sensor and feature evaluation (discrimination), cluster formation and distributed smart fusion in a network of both simulated radar sensors and a network of real disparate sensors and sensor nodes that are spatially distributed. First, in section 18.3.1, the results obtained using a simple simulated network of radar sensors is provided for the purposes of proving the concepts. In section 18.3.2 however, experimental results obtained by using a real DSN of disparate sensors is provided.

### 18.3.1 Simulated network of radar sensors

This network of sensors is used for tracking multiple targets. Each sensor node has a local and global Kalman filter based target trackers. These target trackers estimate the target states - position and velocity in Cartesian co-ordinate system. The local tracker uses the local radar sensor measurements to estimate the state estimates while the global tracker fuses target states obtained from other sensors if it is consistent and improves the accuracy of the target tracks.

For the purposes of testing the proposed algorithms of this chapter, a network of three radar sensors and a single moving target with constant velocity are considered. Two sensors are considered as

good and one as bad. A sensor is defined as bad if its measurements were corrupted with high noise (for example SNR = -6 dB) or is biased. In the first set of examples the SNR of a good sensor is set to be 10 dB.

In the case of simulation of a biased sensor, the bias was introduced as the addition of a random number to the true position of a target. The bias was introduced this way because the biases in azimuth and range associated with a radar sensor translate into measured target position that is different from the true target position. In addition, in our simulations, we assume that the sensors are measuring the target's position in the Cartesian co-ordinate system instead of the polar co-ordinate system. The amount of bias was varied by multiplying the random number by a constant  $k$  i.e., measured position = (true position +  $k \cdot \text{randn}$ ) + measurement noise.

First, the minmax entropy principle was applied to find the missing information, the appropriate sensor was probed to obtain that information, then the consistency measure – within class entropy was applied to check whether the new sensor type and the information obtained from that particular sensor is consistent with the other sensors.

In the following two figures, within class entropy is plotted for features discovered from two unbiased sensors and, one biased and one unbiased sensor. The measurement noise level was kept the same for all three sensors. However, the bias  $k$  was set to 1.0 in Figure 18.1 and was set to 2 in Figure 18.2. The within class entropy was computed for different iterations using the definition provided in the previous section. The probability values needed in this computation were estimated using the histogram approach which is elaborated below. From these two figures, it can be seen that the within class entropy of two unbiased sensors is greater than the within class entropy of one biased and one unbiased sensors. This indicates that the within class entropy can be used as a measure to discriminate between sensors or to assess the quality of sensors (to select sensors).

Next, the conditional entropy and mutual information measures described in the previous section are used to make sure the estimated features obtained from the selected sensors indeed aid in the decision process.

For this, the target states that were estimated from the measurements of a simulated radar at each sensor node using the local Kalman filter algorithm is used as feature sets. The estimated target states at each sensor node were transmitted to other nodes. For this simulation, only estimated position was considered for simplicity.

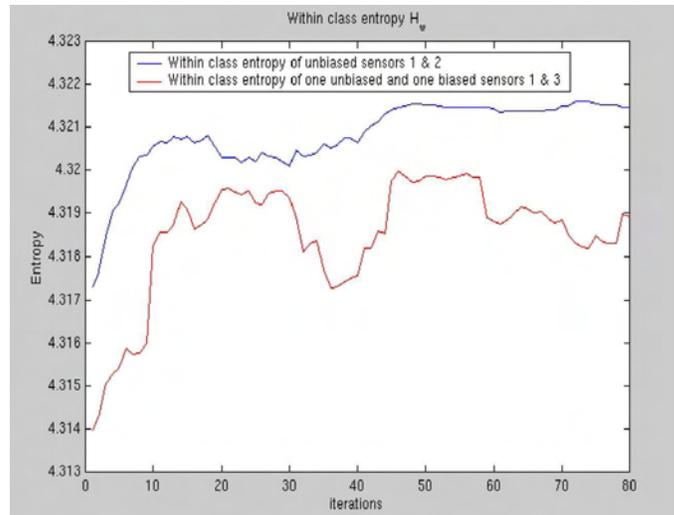


Figure 18.1: The plot of within class entropy of sensors 1 & 2 (unbiased sensors) and, 1 (unbiased) and 3 (biased). Bias constant  $k = 1$

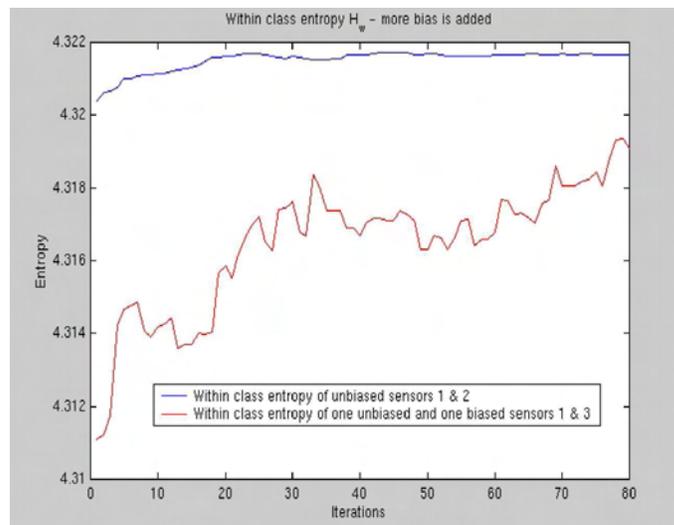


Figure 18.2: The plot of within class entropy of sensors 1 & 2 (unbiased sensors) and, 1 (unbiased) and 3 (biased). Bias constant  $k = 2$

We considered the estimated state vector as the feature set here. Since the goal of this simulation is proof of concept, the feature discrimination algorithm was implemented at sensor node 1 with the assumption it is a good sensor. Let the state estimate outputs of this node be  $A_g$ . Let the state estimate outputs of a second sensor correspond to  $B_g$  and a third sensor correspond to  $B_b$ .

For the computation of entropy, the probability values are needed as seen from the equation above. To obtain these values, ideally, one would need probability distribution functions (pdfs). However, in practice it is hard to obtain closed form pdfs. In the absence of knowledge of actual pdfs it is a general practice to estimate them by using histograms [11]. Researchers in signal and image processing use this technique most commonly [13]. Another practical solution to estimate the probability and conditional probabilities is by using the counting or frequency approach [12]. However, it is well known that the estimates of probabilities and conditional probabilities are more accurate if they are estimated by using the pdfs that are approximated from the histograms. Therefore, we use the histogram approach here. In order to obtain the histograms, initially, we need some data (features) to know how it is distributed. For this purpose, it was assumed that initially  $N$  state estimate vectors were accumulated at each sensor node and this accumulated vector was transmitted to other nodes. Note also that the accuracy of probability estimates using the histogram approach depends on the amount of accumulated (training) data. Also for non-stationary features, it depends on how often the histograms are updated. In practice, since the training data is limited we have set  $N$  to 10 in this simulation. To take care of the non-stationarity of the features, initially, we wait till  $N$  estimates are obtained at each node. From then on we update the histograms every time instant using the new state estimate and previous nine state estimates. At each time instant we discard the oldest feature (oldest state estimate).

To get the probability of occurrence of each feature vector, first the histogram was computed. For this, bin size  $N_{\text{bin}}$  of 5 was used. The center point of each bin was chosen based on the minimum and maximum feature values. In this simulation the bin centers were set as:

$$\min(\text{feature values}) + (0 : N_{\text{bin}} - 1) \cdot \frac{\max(\text{feature values}) - \min(\text{feature values})}{N_{\text{bin}}} \quad (18.12)$$

Since the histogram provides the number of elements in a given bin, it is possible to compute the probabilities from the histogram. In particular it is computed as:

$$\frac{\text{Number of elements in a particular bin}}{\text{Total number of elements}}$$

Hence, from these histograms, probabilities were computed. Similarly, conditional probabilities of  $p(A_g|B_g)$  and  $p(A_g|B_b)$  were computed from the conditional histograms and these conditional probabilities are plotted in Figures 18.3 and 18.4, respectively.

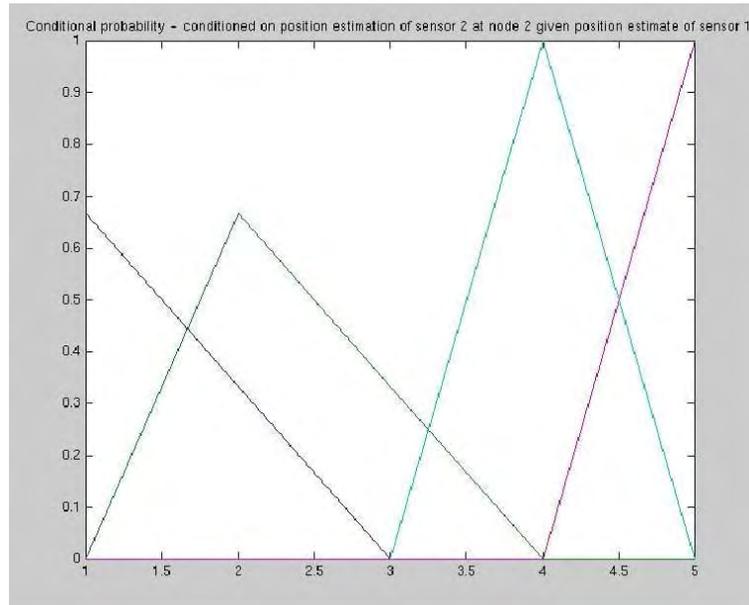


Figure 18.3: Conditional probability of position estimates of sensor 2 at node 2 given position estimates of sensor 1 at node 1

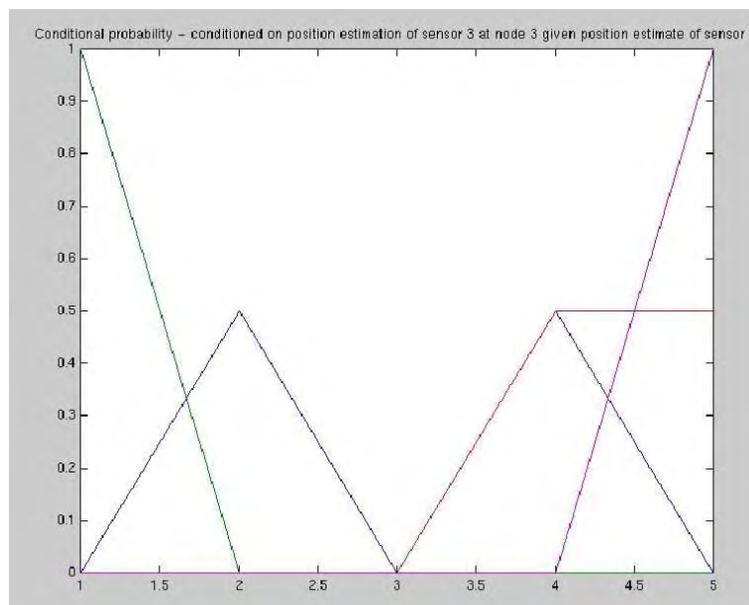


Figure 18.4: Conditional probability of position estimates of sensor 3 at node 3 given position estimates of sensor 1 at node 1

Each colored line in these two plots represents one conditional probability distribution function. Note that both  $A$  and  $B$  are vectors and there would be one pdf for each member of set  $A$ . Since we have chosen bin size as 5 there would be 5 members in set  $A$  and hence, there are 5 subplots in Figures 18.3 and 18.4.

Using these probabilities, conditional entropies  $H(A_g|B_g)$  and  $H(A_g|B_b)$ , and mutual information  $I(A_g, B_g)$  and  $I(A_g, B_b)$  were computed using the equations mentioned above for one set of features from sensor at node 2 and node 3. After this kind of initial computation of probabilities, conditional entropy and mutual information, whenever a sensor estimates a new feature it is replaced by the oldest feature in the feature set and transmitted to other sensors. Subsequently, histograms, probabilities, conditional entropy and mutual information were computed using this updated feature set. This would take care of the non-stationarities of features. Thus each new feature can be verified to make sure it is relevant in terms of aiding in the decision process (e.g., track accuracy) and it is obtained from a good sensor. Therefore, this technique is dynamic in nature.

#### 18.3.1.1 Versatility of the algorithm

To verify the versatility of this algorithm we considered a different feature sets namely, the sensor measurements itself instead of the position estimates and the first difference in position estimates. We performed similar simulation that is described above using these two types of feature sets and the associated histograms for the probability, entropy and mutual information computations. In these two cases also we always obtained  $I(A_g, B_g) > I(A_g, B_b)$  for all the 100 runs of Monte Carlo simulations.

#### 18.3.1.2 Sensitivity of the algorithm for sensor discrimination

Next, noise level at sensor 2 and 3 were varied to determine the sensitivity of the sensor discrimination algorithm. The SNR at sensor 1 was fixed at 10 dB. The algorithm was able to discriminate between good and bad sensor 100 % of the time when the noise level at sensor 2 is 8 dB and at sensor 3 is 3 dB. The algorithm was able to discriminate about 80 % of the time if the noise level at sensor 3 is 5 dB when the noise level at sensor 2 is fixed at 8 dB. If the noise level at both sensor 1 and 2 is 10 dB then the algorithm was able to discriminate 100 % of the time when the noise level at sensor 3 is 5 dB. However, when the noise level at sensor 3 was changed to 7 dB, the percentage of correct discrimination was dropped to 82 %. *Therefore, if the minimum difference between the noise level at sensor 2 and 3 is 5 dB then the discrimination accuracy is 100 %.* If the noise level at both sensor 2 and 3 is close (a difference of 1 dB) then the algorithm cannot discriminate as expected.

### 18.3.1.3 Mutual information versus track accuracy

To check indeed when mutual information metric is used to evaluate the information gain by observing the estimated missing features (information) and it aids in the improvement of the accuracy of decision (e.g., track accuracy), the following experiment was conducted.

As before, mutual information  $I(A_g, B_g)$  and  $I(A_g, B_b)$  was computed using measurements as feature set. If  $I(A_g, B_g) > I(A_g, B_b)$  then the state estimates from the good sensor was fused with sensor 1 using the global Kalman filter algorithm and the DS<sub>m</sub> combination rule that is described in section 18.2.5. The position estimation error was computed by comparing the fused state estimate with the true position. To compare the track accuracies, the state estimates from the bad sensor and good sensor were also fused. The position estimation error was then computed the same way as explained above.

In Figure 18.5, the position estimation error using the fused state estimates of sensor 1 & a good sensor (blue plot) and sensor 1 & a bad sensor (red plot) are plotted. From this figure, it can be seen that the track accuracy after fusing state estimates from good sensors (1 & 2) is much better than fusing state estimates from a good sensor and a bad sensor (1 & 3). This implies that better mutual information correlates to better track accuracy.

In Figure 18.6, the position error is plotted for the case when the noise level at sensor 2 and 3 differs by 5 dB. In this case also it can be seen that the track accuracy is better when the state estimates from good sensors is fused as compared to the track accuracy of fused state estimates of a good sensor and a bad sensor.

We then form a cluster of sensors that are consistent and apply the mutual information metric. We have shown above that by fusing information from sensors when the mutual information increases, the decision accuracy improves. We transmit the fused decision (which requires much lower bandwidth compared to the transmission of decision of each sensor to every other in the network) to other clusters of sensors and thus reduce the communication bandwidth requirement.

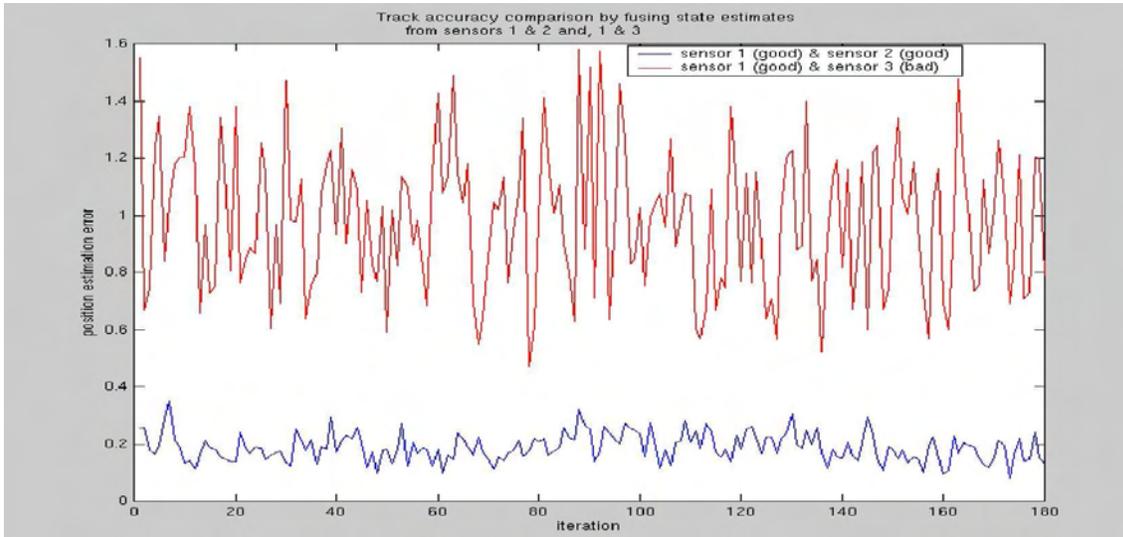


Figure 18.5: Track accuracy comparison - Noise level at sensor 1 and 2 = 10dB and at sensor 3=0dB

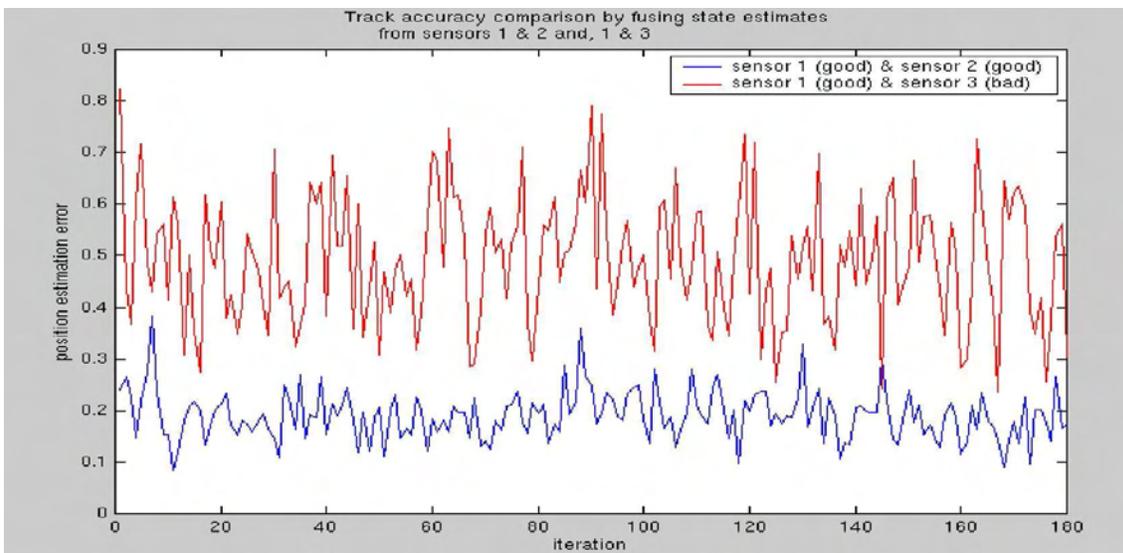


Figure 18.6: Track accuracy comparison - Noise level at sensor 1 and 2 = 10dB and at sensor 3=0dB

### 18.3.2 A real network of spatially DSN with disparate sensors

The proposed algorithms described in section 18.2 was implemented on sensor nodes that consists of multiple sensors, a communication radio and a Sharc processor. These sensor nodes were distributed in a rough terrain such as a desert. This network was used in detecting, tracking and classifying targets. Even though we verified the algorithms that estimate the missing information, sensor selection, sensor and feature assessment in this network of sensor node, in the following subsections, we are concentrating on the value of information based smart fusion that is described in sections 18.2.4 and 18.2.5 since the experimental results for the other algorithms are provided in the last section. We provide the experimental details and the results. We begin with the review of detection and classification algorithms that were used in this context.

#### 18.3.2.1 Review of algorithms used to check the value of information based smart fusion

The metrics described in section 2.4 are used to measure the value of information obtained from other sources such as multiple sensors on a single node and from the neighboring nodes in the context of target detection and classification. For target detection, energy based detector was used and for classification, maximum likelihood based classifier was used. As mentioned before the value of information is in terms of improvement in the decision accuracy which corresponds to classification accuracy for a classifier and detection accuracy or probability of detection for a detector. Note that in this study, we did not develop a classifier or a detector; however, used those developed by others since the goal of this part of the study is to develop measures of value of information and verify them in terms of improvement in decision accuracy when they were used to make a decision of whether to fuse information obtained from the other source or not. In the following two sections we review the classifier and the detector that were used in this study.

**18.3.2.1.1 Maximum likelihood based classifier** The classifier we used for the verification of measures of value of information in terms of improving the decision accuracy is a maximum likelihood based classifier developed by the University of Wisconsin [16] as part of DARPA's sensor information technology (SensIT) program. For a given training features and target labels a Gaussian mixture model is determined during the training phase of the classifier. During testing the distance between the test feature vector and  $i$ th class Gaussian mixture is computed. This corresponds to negative log likelihood. Then a priori probability is used to obtain the maximum a posterior classification. The features' set that is used here consists of twenty features from the power spectral density. This is computed using 1024 FFT. The feature set is collected by summing up the values over equal length segments of the power spectrum. For the acoustic and seismic sensors the maximum frequency used was 1000 and 200 Hz, respectively.

**18.3.2.1.2 Energy based detector** An energy based detector is also used for the verification of improvement in decision accuracy when the value of information based fusion architecture is used. This detector is developed by BAE, Austin [3]; also as part of the SensIT program. A brief description of this detector is provided below.

For every block of a given signal the energy of the down sampled version of the power spectral density is computed. For the computation of the power spectral density, 1024 point FFT is used. This energy is compared with a threshold value. Whenever the energy is above the threshold it was declared that the target was detected. The threshold value is adaptively changed based on the background energy.

### 18.3.2.2 Experimental details

The above described classifier and detector, and measures of value of information and the fusion algorithm which uses these measures while deciding when to and when not to fuse information were implemented and were tested using real data that was obtained by distributing sensor nodes along the east-west and south-north road at Twentynine Palms, CA during one of the field tests (SITEX'02) as shown in Figure 18.7. These sensor nodes are manufactured by Sensoria. On each sensor node, three sensors - acoustic, seismic and IR sensors, a four channel data acquisition board and a processing board are available. These nodes also have communication capabilities. For more details on the sensor node, refer to [14].



Figure 18.7: Sensor node distribution at Twenty nine Palms, CA

Three vehicles – AAV, Dragon Wagon (DW) and HMMWV were driven along the east-west and north-south road as shown in Figure 18.7 while conducting the experiments. In this figure, nodes placements are also provided. Totally twenty four nodes were considered in our experiments. We used both seismic and acoustic data from these nodes when it is appropriate. In the next section, the classification experimental details and the results are provided and in section 18.3.2.2.2 the detection experiments and the results are provided. In both these sections experimental details and results are provided with and without value of information based fusion technique that was developed in this study.

**18.3.2.2.1 Classification experiments** First, acoustic data from each node is considered. The maximum likelihood classifier is trained using only acoustic data from individual nodes. The challenges in the classification experiments are threefold: 1) when to reject a source of data, 2) when to propagate data between sequential nodes, and 3) when to share individual sensor data within the same node. Using only acoustic data, we investigated the effectiveness of the four measures of value of information outlined in Section 18.2.4 - mutual information, Euclidean distance, correlation, and Kullback-Liebler distance.

In addition, we investigated two methods of using these measures. When evaluating the effectiveness of fusing two sources of data, is it better to compare the two sources with each other or with the stored training data? To answer this question, we devised several similarity measures to measure the closeness of two data sources. We calculated these measures between data at all sequential nodes. Then for each similarity measure, we computed its correlation with correct classification performance at each node. We call this the performance correlation. The average performance correlation over all nodes for each class of data using previous node similarity measures is shown in Figure 18.8. Next, we calculated the same similarity measures between the data at each node and the data stored in the training sets. Again, for each similarity measure, we computed its correlation with correct classification performance at each node.

The average performance correlation over all nodes for each class of data using training set similarity measures is shown in Figure 18.9.

Inspection of Figures 18.8 and 18.9 show that the similarity measures Euclidean distance and correlation are more closely aligned with correct classification performance than either mutual information or Kullback-Liebler distance. In practice, however, we found that the Euclidean distance outperformed correlation as the determining factor in fusion decisions. Furthermore, comparing Figures 18.8 and 18.9 shows that using the training set for similarity measures is more effective than using the data from the previous node in the network. We found this to be true in practice as well. Subsequent work with the seismic data echoed the findings of the acoustic data. Note that even though we use the training data to make the fusion decision, we perform the actual data fusion with current and previous node data.

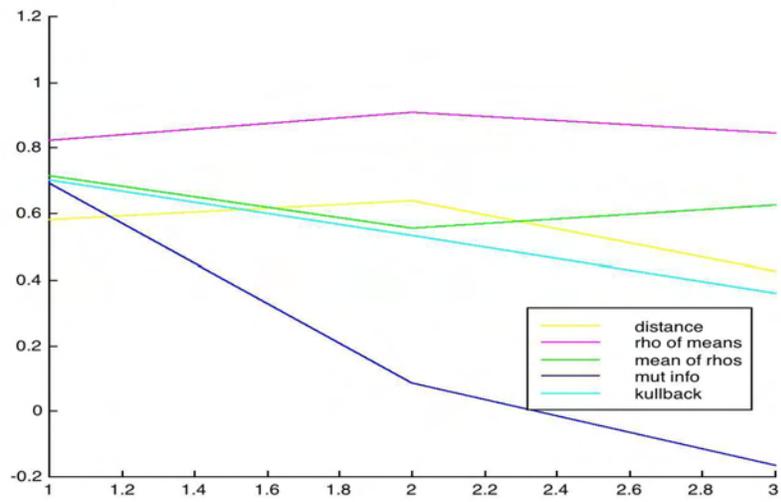


Figure 18.8: Performance correlation of previous node data

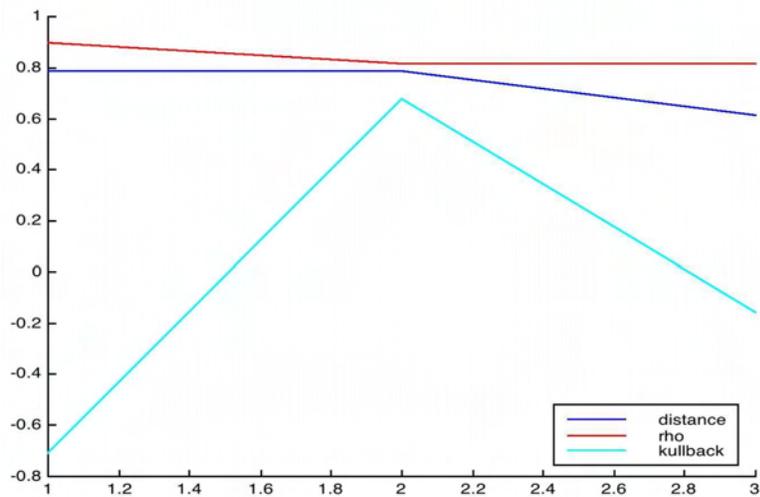


Figure 18.9: Performance correlation of training class data

**Rejection of bad data** Sometimes one node or one sensor can have bad data, in which case we prefer to reject this data rather than classify with poor results. The feature discrimination algorithm is used for this. By rejecting the data, we did not fuse it with any other data, pass it on to any other node, nor even compute a classification at that source. Our method resulted in the rejection of several sources of bad data, thus improving the overall classification results as shown in Figures 18.10 and 18.11.

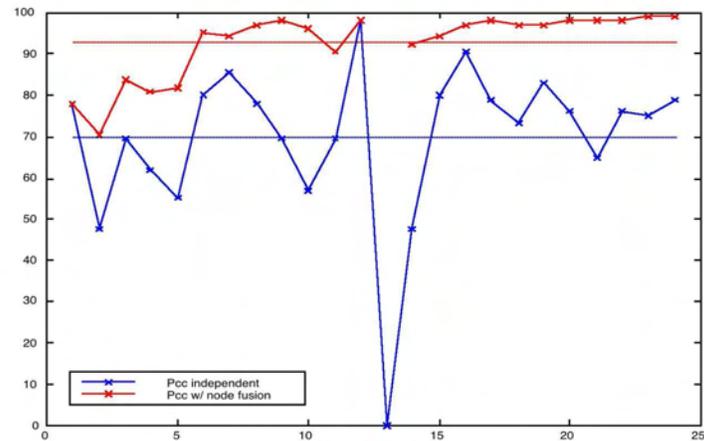


Figure 18.10: Performance of node fusion for the AAV with acoustic sensor data

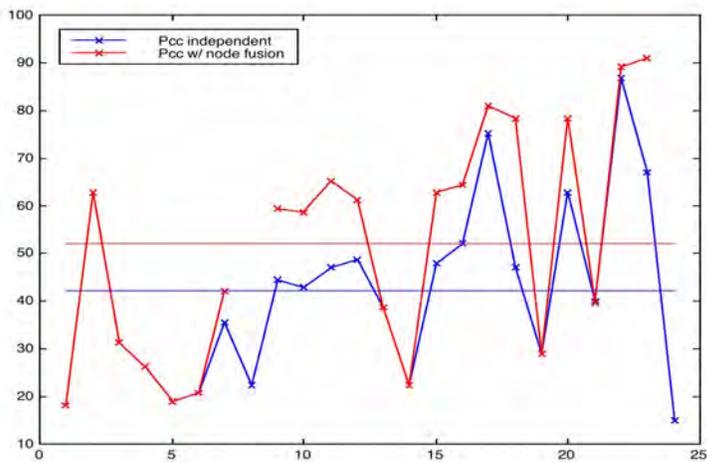


Figure 18.11: Performance of node fusion for the DW with seismic sensor data

**Node to node fusion** The fusion decision can be made with a threshold, i.e. if the distance between two features sets is below some value, then fuse the two feature sets. The threshold value can be predetermined off-line or adaptive. We sidestep the threshold issue, however, by basing the fusion decision on relative distances. To do so, we initially assume the current node belonged to the same class (aka the target class) as the previous node and employ the following definitions. Let  $x_n$  be the mean vector of the current node data. Let  $x_{nf}$  be the mean vector of the fused data at the current node. Let  $x_{c_1}$  be the mean vector of the target training class data. Let  $x_{c_2}, x_{c_3}$  be the mean vectors of the remaining training classes. A Euclidean distance ratio is defined as:

$$r_{dist} = d_{c_1} / \min(d_{c_2}, d_{c_3}) \quad (18.13)$$

where  $d_{c_i}$  is the Euclidean distance (18.8) between  $x_n$  and  $x_{c_i}$ . We then use the following pseudocode to make our fusion decisions.

```

if ( $r_{dist} \leq 1.0$ )
    fuse_4class = 1; fuse_4carry = 1;
    class_ind = classify  $x_n$ ;
    if (class_ind  $\geq$  70%) check class_fuse;
end
else
    fuse_4class = 0; fuse_4carry = 0;
    if ( $(d_{c_1} \leq 3s_{e1}) \& (d_{c_2} \leq 3s_{e2}) \& (d_{c_3} \leq 3s_{e3})$ )
        class_ind = classify  $x_n$ ;
        if (class_ind == target class) fuse_4class = 1;
        if (class_ind  $\geq$  70%)
            fuse_4carry = 1;
            class_fuse = classify  $x_{nf}$ ;
        if (class_ind > class_fuse)
            class_fuse = class_ind;
        end
    end
end
else
    reject this data;
end
end

```

There are two outcomes to the fusion decision. First we decide whether or not to fuse the data at the current node. If the current node has bad data, fusion can pull up the performance, however, we may not want to carry the bad data forward to the next node (the second fusion decision outcome). `fuse_4class` is a flag indicating whether or not to fuse for the current classification. `fuse_4carry` is a flag indicating whether or not to include data from the current node in the fused data that is carried forward. Based on this decision, the fusion of classification decision is achieved by applying the fusion algorithm described in section 18.2.5. In Figures 18.10 and 18.11 we show the correct classification improvement gained by fusing from node to node for the acoustic and seismic sensors, respectively. For the acoustic sensor we show classification results from the AAV data, while using DW data for the seismic sensor results. In the case of the acoustic data, the mean correct classification performance across all nodes increases from 70% for independent operation to 93% with node to node fusion across the network. Similarly, the seismic correct classification performance increases from 42% to 52%.

**Fusion between sensors** After fusion from node to node of the individual sensors, we look at the benefit of fusing the acoustic and seismic sensor data at the same node. To do so, we employ the following definitions. Let  $r_{dist}$  be defined as in (18.13) but with the new data types (a - acoustic, s - seismic, and as - a concatenated acoustic/seismic vector). Let  $x_a$  be the mean vector of the current node acoustic data after fusion from node to node. Let  $x_s$  be the mean vector of the current node seismic data after fusion from node to node. Let  $x_{as} = x_a$  concatenated with  $x_s$  (dumb fusion). Let  $x_{asf}$  = smart fusion of  $x_a$  with  $x_s$ . Let  $x_{in}$  be the data input to the classifier. Now, we employ two steps in the sensor fusion process as shown in the pseudocode below. In this case also for the fusion of features from two independent sources such as acoustic and seismic, DS<sub>m</sub> based technique described in section 18.2.5 is applied. First we employ a smart sensor fusion routine:

```

indx = min( $r_{adist}$ ,  $r_{sdist}$ ,  $r_{asdist}$ )
if (indx == 1)  $x_{in} = x_a$ ;
elseif (indx == 2)  $x_{in} = x_s$ ;
elseif (indx == 3)  $x_{in} = x_{as}$ ;
end

```

Next, we employ a final fusion routine:

```

class_acst = classify  $x_a$ ;
class_seis = classify  $x_s$ ;
class_as_dumb = classify  $x_{as}$ ;
class_as_smart = classify  $x_{asf}$ ;
if { (class_acst >= 70%) | (class_seis >= 70%) | (class_as_ind >= 70%) }
    class_final_fuse = max (class_acst, class_seis, class_as_dumb, class_as_smart)
end

```

Figure 18.12 shows the results of fusion at each stage in the form of a bar plot. The classification performance is averaged over all the nodes for each vehicle class. The correct classification performance improves at each stage of fusion processing as shown in Table 18.1. The results indicate that the fusion based on value of information helps in improving the decision accuracy at each node significantly.

	AAV	DW	HMMV
Acoustic independent	70%	58%	46%
Seismic independent	72%	42%	24%
Acoustic fusion	93%	80%	69%
Seismic fusion	93%	52%	31%
Acoustic & seismic, independent	76%	55%	58%
Acoustic & seismic with fusion	95%	90%	77%

Table 18.1: Summary of classification performance

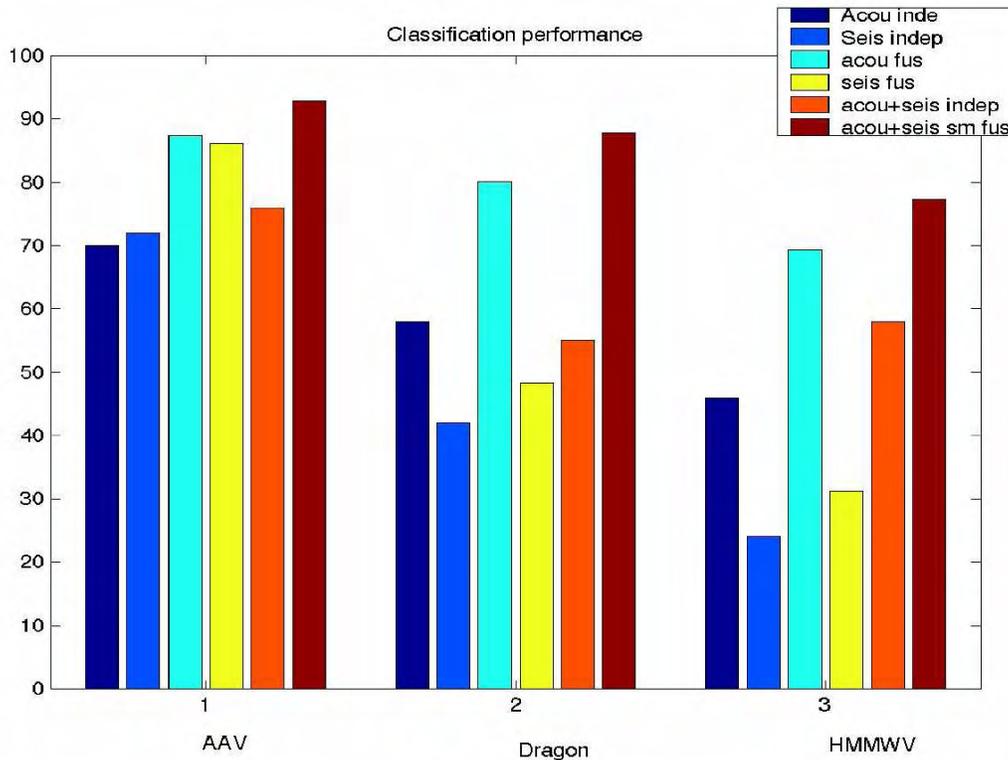


Figure 18.12: Average correct classification performance at each step in the fusion process

**18.3.2.2.2 Detection experiments** For the detection experiments also both acoustic and seismic data were considered. First, only acoustic data from individual nodes were used. A threshold value was initially set which was varied adaptively based on the background energy. The power spectral density of acoustic data was computed using 1024 point FFT and it was downsampled by 8. The energy of the downsampled version of the power spectral density was computed. This energy was compared with the threshold value. If the energy was above the threshold value, it was decided that the target was detected. The time of detection and the confidence on detection were also calculated. The detection and time of detection were compared with the ground truth. If the target was detected when it is supposed to be and if the time of detection is within the region of interest then it was counted towards calculating the probability of detection. If the detection time is outside the region of interest (missed detection) and if a target was detected when it should not have been (false alarm) it was counted towards computing the probability of false alarm. The probability of detection and false alarm using only acoustic data from individual nodes without any fusion for AAV, DW and HMMWV are: 0.8824, 0.8677, 0.8382 and 0.1176, 0.1323, 0.1618, respectively. Similarly, the probability of detection and false alarm using only seismic data from individual nodes without any fusion for AAV, DW and HMMWV are: 0.8030, 0.7910, 0.5735 and 0.1970, 0.2090, 0.4265, respectively.

Next, the mutual information based value of information measure was used on the energy of power spectral density to make a decision of fusing data between sensors - acoustic and seismic on each individual node. The detector was tested using the fused data on each node. The probability of detection and false alarm were computed as described above. The probability of detection of this intelligently fused data for AAV, DW and HMMWV is: 0.9394, 0.9105 and 0.8529, respectively. The probability of false alarm is not provided here because it is equal to  $1 - \text{probability of detection}$  since both false alarm and missed detections are combined together. These results are summarized in Figure 18.13 in the form of a bar graph. From this, it can be seen that the intelligent sensor data fusion based on value of information and DSMT significantly improves the detection accuracy. This type of fusion especially helps in difficult data as in the case of HMMWV.

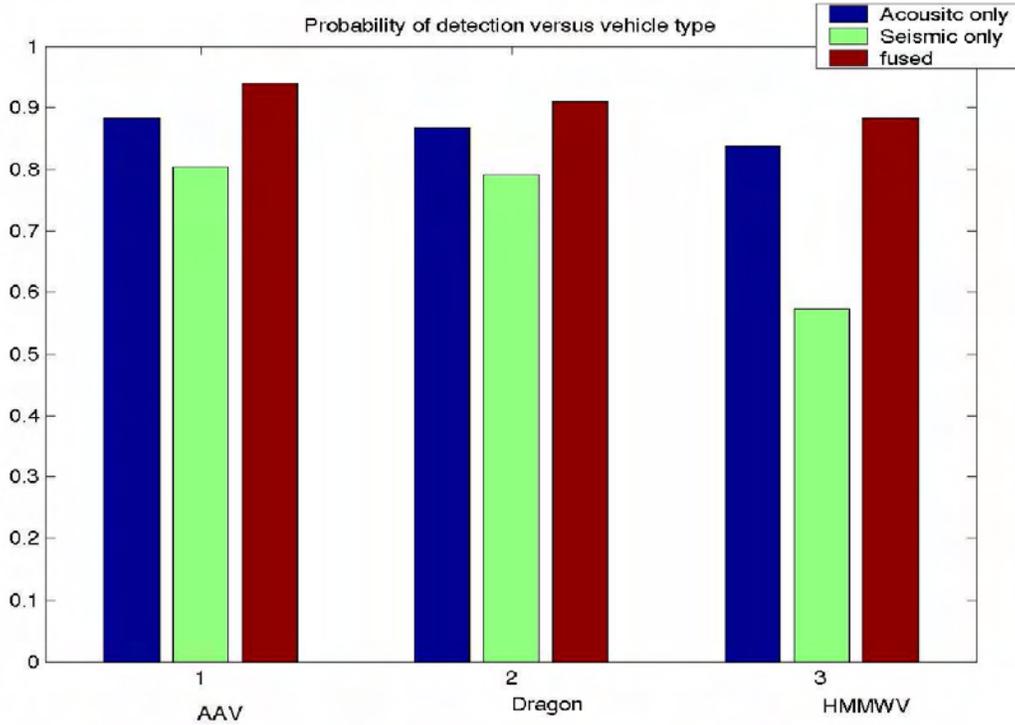


Figure 18.13: Performance of a detector

## 18.4 Conclusions

In this chapter, we have described how minmax entropy principle can be used in feature (missing information) discovery and the type of sensor (information type) from which this missing information can be obtained. Further, a consistency measure is defined and it has been shown that this measure can be used in discriminating or assessing the quality of sensors. Next, conditional entropy and mutual information measures are defined and it has been shown that these two measures can be used in making sure that the estimated missing information or new feature set indeed help in gaining information and aid in decision process. Further more, we have introduced several measures for value of information. We have used these measures in deciding when to fuse information. For the fusion we have developed an algorithm using DSmT. We have proven the concept of all the measures and fusion by first considering a simulated network of radar sensors and then by considering a real network of spatially distributed sensor nodes which have multiple sensors on each sensor node. The experimental results indicate that (a) the minmax entropy principle can be used in estimating the missing information and information type and it can be used in the cluster formation; (b) the constancy measure based on within class entropy can be used in sensor discrimination; (c) the mutual information can be used in feature quality assessment and in evaluating the value of information; (d) the measures of value of information helps in smart fusion; (e) the distributed smart fusion significantly improves the decision accuracy. All these measures help in probing (awakening) the required sensor for the required missing information, only transmitting the valuable information when and where it is needed and fusing only valuable information. Thus, power and, computing and communication resources can be efficiently utilized.

## 18.5 References

- [1] Bar-Shalom Y., Fortmann T.E., *Tracking and data association*, Academic press, San Diego, 1988.
- [2] Battti R., *Using mutual information for selecting features in supervised neural net learning*, IEEE Trans. On Neural Network, Vol. 5, No. 4, July 1994, pp. 537-550.
- [3] Beck S., *Energy based detector*, SensIT PI meeting, Jan 15-17, 2002, Santa Fe, New Mexico.
- [4] Darbellay G.A., Vajda I., *Estimation of the information by an adaptive partitioning of the observation space*, IEEE Transactions on Information Theory, vol. 45, no. 4, pp. 1315-1321, May 1999.
- [5] Dezert J., Smarandache F., *Combination of beliefs on hybrid DSm models*, Advances and Applications of DSmT for Information Fusion, American Research Press, Rehoboth, June 2004, Chap. 4.
- [6] Kadambe S., *Information theoretic based sensor discrimination for information fusion and cluster formation in a network of distributed sensors*, in Proc. of 4th International Conference on Information fusion, Montréal, Québec, Canada, August 7-10, 2001, pp. ThC1-19-ThC1-25.

- [7] Kadambe S., *Feature discovery and sensor discrimination in a network of distributed sensors for target tracking*, in Proc. of IEEE workshop on Statistical signal processing, Singapore, August 6-8, 2001, pp. 126-129.
- [8] Kadambe S., Daniell C., *Sensor/Data Fusion Based on Value of Information*, in Proc. of 6th International Conference on Information Fusion, Fusion 2003, Cairns, Australia, July 2003, pp. 25-32.
- [9] Kapur J. N., *Measures of information and their applications*, John Wiley, Eastern Limited, 1994.
- [10] Manyika J., Durrant-Whyte H., *Data fusion and sensor management: An information theoretic approach*, Prentice Hall, 1994.
- [11] Papoulis A., *Probability, Random variables and Stochastic Processes*, Second edition, McGraw Hill, 1984, pp. 500-567.
- [12] Press W.H., Teukolsky S. A., Vetterling W. T., Flannery B. P. , *Numerical Recipes in C*, Cambridge University Press, 1992, pp. 632-635.
- [13] Rabiner L. R., Juang B-H. , *Fundamentals of Speech Recognition*, Prentice Hall, 1993, Chapter 6.
- [14] <http://www.sensoria.com>
- [15] Thomopoulos S. C. A. , *Sensor selectivity and intelligent data fusion*, Proc. Of the IEEE MIT'94, October 2-5, 1994, Las Vegas, NV, pp. 529-537.
- [16] Hen Wu Y., *Maximum likelihood based classifier*, SensIT PI meeting, Jan 15-17, 2002, Santa Fe, New Mexico.