Learning the Quality of Sensor Data in Distributed Decision Fusion

Bin Yu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
byu@cs.cmu.edu

Katia Sycara
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
katia@cs.cmu.edu

Abstract - The problem of decision fusion has been studied for distributed sensor systems in the past two decades. Various techniques have been developed for either binary or multiple hypotheses decision fusion. However, most of them do not address the challenges that come with the changing quality of sensor data. In this paper we investigate adaptive decision fusion rules for multiple hypotheses within the framework of Dempster-Shafer theory. We provide a novel learning algorithm for determining the quality of sensor data in the fusion process. In our approach each sensor actively learns the quality of information from different sensors and updates their reliabilities using the weighted majority technique. Several examples are provided to show the effectiveness of our approach.

Keywords: distributed sensor systems, decision fusion, quality of information, Dempster-Shafer theory.

1 Introduction

Distributed sensor systems have recently received significant attention in the areas of networking and multi-agent systems [1, 2, 3, 4]. In this paper we are particularly interested in autonomous mobile sensors, such as robots and UAVs (unmanned aerial vehicles). These networked mobile sensors play important roles in military and civilian operations [5]. The motivating examples include disaster search and rescue, security surveillance, and battlefield operations, where hundreds of robots or UAVs are deployed in a partially unknown environment to search for victims or targets [6, 7].

One of the fundamental problems in distributed autonomous sensor systems is data fusion, as the raw data from each sensor is usually uncertain and noisy and cannot be used directly for plan instantiation and team coordination [8]. Over the past two decades, a large number of approaches to multisensor data fusion have been developed in the community of information fusion [9, 10]. Generally, data fusion could happen from raw data level such as images to the decision level. Here we focus on decision level data fusion, where a sensor seeks to fuse decisions made by other sensors. A decision for a target’s identity can be either single or multiple hypotheses. In the literature, a single hypothesis decision is also called a hard decision, while a multiple hypotheses one is known as a soft decision.

Decision-level data fusion has been studied for a long time. For example, optimal decision fusion for multisensor detection systems was studied in 1980s by Chair and Varshney [11] and Thomopoulos et al. [12, 13]. More recently, Niu et al studied the problem in a distributed sensor system, where a decision fusion rule was developed based on the total number of detections made by local sensors [14]. These approaches mainly focus on binary hypothesis testing in multisensor detection systems based on statistical inference techniques such as Neyman-Pearson test. Although the concept of sensor reliability has been introduced by Thomopoulos et al., they do not provide any formalism to assess the changing quality of sensor information for data flows [15].

We consider multiple hypotheses decision fusion in distributed sensor systems, where the decision from each sensor is not binary but instead a multiple declaration of identity for a target. We present a novel approach to adaptive decision fusion using Dempster-Shafer theory. In our approach, a sensor forwards the decision (also known as data or report) to one of its neighbors once it detects a new target. Any sensor that fuses all relevant data for a target can stop the data propagation if the confidence of fused data meets a threshold. Different from binary hypothesis testing, the rules for multiple hypotheses decision fusion are adaptive. The number of sensor reports that needs to be fused is dynamically changing and relies on the threshold level and the specific target type.

Moreover, we provide a novel learning algorithm for determining the quality of sensor data in the fusion process. The idea is that different sensors can generate different qualities of information. Therefore, it would be useful for each sensor to be able to differentiate the sources of higher quality. Specifically, each sensor actively learns the quality of information from different sensors and updates their reliabilities using the weighted majority technique [16]. The effect of information quality on data flows is quantified via the notion of conflicts in Dempster-Shafer theory. A simple application of our approach is illustrated in a military context, where a team of UAVs are deployed for target detection and classification in the battlefield.
The rest of this paper is organized as follows. Section 2 briefly describes the background of distributed sensor systems. Section 3 investigates the approach to decision fusion in distributed sensor systems. Section 4 illustrates a simple application of our approach using the simulated testbed OTBSAF. Section 5 concludes this paper and presents some directions for future research.

2 Distributed Sensor Systems

One example of distributed sensor systems is UAVs in military applications. In a military context, various radars, e.g., SAR (Synthetic Aperture Radar), EO (Electro-Optical radar), and GMTI (Ground Moving Target Indicator), are mounted on a number of platforms. These platforms such as UAVs are deployed in the battlefield as a distributed autonomous sensor system for target detection, tracking, classification, attack, and damage assessment.

We assume that mobile sensors communicate on a point-to-point basis [17]. A network of these mobile sensors is modeled as a connected undirected graph \( G = (V, E) \), where \( V = \{a, b, \ldots\} \) is a set of sensor nodes and \( E \) consists of edges between any two nodes \( a \) and \( b \) that can communicate directly. Specifically, \( N(a) \) is the set of \( a \)'s neighbors and \( b \in N(a) \) is any neighbor of node \( a \).

Moreover, we assume there are multiple stationary targets \( T_1, T_2, \ldots, T_M \) on the ground. Given a target \( T_K \), a SAR sensor will return a list of candidate target types with different confidence levels. A target \( T_K \) may be detected by multiple UAVs at the same time, but each UAV cannot initialize its plans of target engagement based on its own raw sensor data. The reason is that the raw sensor data from a UAV is uncertain and noisy. Sometimes, a UAV with a SAR sensor may even confuse a friendly target (an M1A1 tank) with an enemy target (a T80 tank).

![Figure 2: An example of data flows in the sensor system, where \( e_i, e_j, \) and \( e_k \) are three relevant events from sensors \( a, d, \) and \( e \) and all of them refer to target \( T_K \).](image)

The process of sensor data delivery can be briefly described as follows.

When sensor \( a \) receives or generates an event \( e_i \) about \( T_K \), it first tries to fuse \( e_i \) with its previous events about target \( T_K \) using data fusion techniques such as Dempster-Shafer theory. If the confidence of fused events meets a threshold, sensor node \( a \) will stop the propagation of event \( e_i \) and become a sink node for relevant events of \( T_K \). Otherwise, sensor \( a \) will choose

- **sender** is the ID of the sensor that detects the target. In practice, the ID could be a random number chosen in a range large enough so that any two sensors will have different IDs. \(^1\)
- **identity** is the decision about the target from the sensor. In our approach, it refers to a multiple hypotheses decision from SAR sensors.
- **location** is the location of the target \( T_K \) being detected, denoted as \( (x_K, y_K, z_K) \).
- **timestamp** is the time that the target is detected by the sensor. The location and timestamp of a target are mainly used in the data association process to determine if two events are referring to the same target. \(^2\)
- **TTL** (time-to-live) is the maximal number of hops allowed for the event propagation in the sensor system.
- **pedigree** is the list of nodes event \( e_i \) has visited, denoted as \( L \). Note that pedigree is used to avoid cycles during event propagation.

![Figure 1: Two lists of candidate target types for a T80 tank from two SAR sensors.](image)

Figure 1 illustrates two lists of candidate target types for a T80 tank from two SAR sensors. One of the SAR sensors confuses the T80 tank with an M1A1 tank in list (b). Hence, the low quality sensor data cannot be used directly for high-level plans and has to be delivered to other nodes for fusion in the system.

If sensor \( a \) detects a target \( T_K \) on the ground, it will generate an event \( e_i \) about target \( T_K \). Formally, an event \( e_i \) about target \( T_K \) can be denoted as a tuple \( \langle \text{sender}, \text{identity}, \text{location}, \text{timestamp}, \text{TTL}, \text{pedigree} \rangle \), where

\[ \text{sender} \]
\[ \text{identity} \]
\[ \text{location} \]
\[ \text{timestamp} \]
\[ \text{TTL} \]
\[ \text{pedigree} \]

\(^1\)One advantage of random numbers is that they scale well to large numbers of autonomous sensors which may dynamically enter and leave the sensor system.

\(^2\)The discussion of distributed data association in sensor systems is outside of the scope of this paper. Readers may refer to [18, 19] for more information.
one of its neighbors to continue the delivery unless its TTL is zero.

Figure 2 shows an example of data flows in the system. The solid lines correspond to directed communication channels between sensor nodes. The arrows in dashed lines represent information flows of relevant events $e_i$, $e_j$, and $e_l$ for target $T_K$. As shown in Figure 2, the three events are fused at node $c$.

3 Adaptive Decision Fusion

Commonly used frameworks for multiple hypothesis decision fusion are Bayesian inference method and Dempster-Shafer theory. Dempster-Shafer theory produces identical results as Bayesian inference method for multisensor decision fusion when the hypotheses about individual target’s identity declarations are singletons and mutually exclusive [20, 21, 22].

However, Bayesian inference method cannot apply to decision fusion in our scenario, as most tactical sensors are incapable of assigning all confidences to each target type. For example, in Figure 1, the non-zero confidence assigned to CLUTTER is not mutually exclusive with other target types. On the contrary, Dempster-Shafer theory allows confidences to be assigned to sets of propositions rather than to just $N$ mutually exclusive propositions.

Therefore, we choose Dempster-Shafer theory as our underlying reasoning framework for decision fusion.

3.1 Dempster-Shafer theory

We now introduce the key concepts of the Dempster-Shafer theory. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of possible vehicle types, where $v_i$, $1 \leq i \leq n$, is the possible type of vehicles a SAR sensor can recognize. In practice, a SAR sensor will report confidence levels for around three dozen entities. The SAR sensor we simulate will report confidence levels for all SAR sensor outputs. The frame of discernment for all SAR sensor outputs is a basic probability assignment [23].

Definition 1 Let $\Theta$ be a frame of discernment. A basic probability assignment (bpa) is a function $m : 2^\Theta \rightarrow [0, 1]$ where (1) $m(\emptyset) = 0$, and (2) $\sum_{A \subseteq \Theta} m(A) = 1$.

In this paper we consider a common frame of discernment for all SAR sensor outputs. The frame of discernment $\Theta$ is $\{T80, T72M, M1, M1A1, M1A2, 256, ZSU23, M977, M35, AV ENGER, HMMWV, SA9\}$. In our scenario, the basic probability assignment can be defined as follows: (1) for any target type $v_i \in V$, $m(\{v_i\}) = c(v_i)$, where $c(v_i)$ is the confidence level of $v_i$ in the table; (2) for any $\hat{A} \subseteq \Theta$ and $\hat{A} \notin V$, $m(\hat{A}) = 0; (3)$ $m(\Theta) = c(\text{CLUTTER})$, where we assign the confidence of CLUTTER to the set of all possible target types.

For a subset $\hat{A}$ of $\Theta$, the belief function $\text{Bel}(\hat{A})$ is defined as the sum of the beliefs committed to the possibilities in $\hat{A}$. For individual members of $\Theta$, $\text{Bel}$ and $m$ are equal. Thus

\[ \text{Bel}(\{T80\}) = m(\{T80\}) \]
\[ \text{Bel}(\{T72M\}) = m(\{T72M\}) \]

A subset $\hat{A}$ of a frame $\Theta$ is called a focal element of a belief function $\text{Bel}$ over $\Theta$ if $m(\hat{A}) > 0$. Given two belief functions over the same frame of discernment but based on distinct bodies of evidence, Dempster’s rule of combination enables us to compute a new belief function based on the combined evidence. For every subset $\hat{A}$ of $\Theta$, Dempster’s rule defines $m_1 \oplus m_2(\hat{A})$ to be the sum of all products of the form $m_1(\{X\})m_2(\{Y\})$, where $X$ and $Y$ run over all subsets whose intersection is $\hat{A}$.

Definition 2 (Dempster’s rule of combination) Let $\text{Bel}_1$ and $\text{Bel}_2$ be belief functions over $\Theta$, with basic probability assignments $m_1$ and $m_2$, and focal elements $A_1, A_2, A_3, B_1, B_2$, respectively. Suppose

\[ \sum_{i,j, A_i \cap B_j = \emptyset} m_1(A_i)m_2(B_j) < 1 \]

Then the function $m : 2^\Theta \rightarrow [0, 1]$ that is defined by $m(\emptyset) = 0$, and

\[ m(\hat{A}) = \frac{\sum_{i,j, A_i \cap B_j = \hat{A}} m_1(A_i)m_2(B_j)}{1 - \sum_{i,j, A_i \cap B_j = \emptyset} m_1(A_i)m_2(B_j)} \]

for all non-empty $\hat{A} \subseteq \Theta$ is a basic probability assignment [23].

\[ \text{Bel}, \text{the belief function given by} \ m, \text{is called the orthogonal sum of} \ \text{Bel}_1 \text{ and} \ \text{Bel}_2. \text{It is written as} \ \text{Bel} = \text{Bel}_1 \oplus \text{Bel}_2. \text{Note that Dempster’s rule of combination is associative and commutative. This means that the processes of combining evidence from multiple sensors are independent of the order in which the sensor outputs are combined.} \]

For any $v_i \in V$, $m(\{v_i\})$ can be simplified as

\[ \frac{m_1(\{v_i\})m_2(\{v_i\}) + m_1(\{v_i\})m_2(\{v_i\})m_2(\{v_i\}) + m_2(\{v_i\})m_1(\{v_i\})}{1 - \sum_{v_j \in V} m_1(\{v_j\})(1 - m_2(\{v_j\}) - m_2(\{v_j\}))} \]

Similarly, we can compute $m(\Theta)$ as

\[ m(\Theta) = \frac{m_1(\Theta)m_2(\Theta)}{1 - \sum_{v_j \in V} m_1(\{v_j\})(1 - m_2(\{v_j\}) - m_2(\{v_j\}))} \]

(3)

When a sensor fuses the data from multiple sensors, it needs to determine if it should stop the data propagation. The key challenge here is how to derive the confidence for a target from the fused sensor data. Next we define a rule which works for the SAR sensor data in our scenario.

Definition 3 Let $m$ be the basic probability assignment for the fused sensor data of target $T_K$, the confidence of $T_K$ is defined as

\[ C(T_K) = \max_{v_i \in V} m(\{v_i\}) + m(\Theta) \]
For any incoming event $e_i$ for target $T_K$, a sensor will stop the propagation of $e_i$ if and only if the confidence of fused sensor data (including $e_i$) is greater than the threshold $\omega$, denoted as $C(T_K) > \omega$.

Once a sensor finds the confidence of target $T_K$ is above the given threshold $\omega$, the sensor will pass the information to the team coordination layer of the sensor system. Specifically, it will create an information token about target $T_K$ when it successfully fused the relevant sensor data for target $T_K$ with high confidence, e.g., $C(T_K) > \omega$. Information tokens are then used for high-level team coordination such as task and resource allocation [5]. For example, the system will autonomously allocate the task of attacking the target to a nearby UAV that possesses the required type of munitions.

### 3.2 Weighted Majority Algorithm

In Section 3.1, we provide a theoretical framework for uncertain sensor data fusion using Dempster-Shafer theory. In practice, the data from different sensors may not be equally reliable due to sensor malfunction or severe weather conditions. Therefore, it is important to take this into consideration during the fusion process and qualify the performance degradation of fusion results.

Next we introduce the weighted majority algorithm (WMA) and its variant continuous weight majority algorithm (WMC). We also provide a novel learning algorithm for determining the quality of sensor data in the fusion process.

The weighted majority algorithm (WMA) deals with how to make an improving series of predictions based on a set of advisors [16]. The first idea is to assign weights to the advisors, e.g., $w_i$, and to make a prediction based on the weighted sum of the ratings provided by them. The second idea is to tune the weights after an unsuccessful prediction so that the relative weight assigned to the successful advisors is increased and the relative weight assigned to the unsuccessful advisors is decreased. WMA aims to design a master algorithm that applies to the combination of evidence without regard to the reasoning on which the individual ratings might be based.

To motivate our approach, we describe a variant of WMA called WMA Continuous (WMC) [16]. WMC allows the predictions of the algorithms to be chosen from the interval $[0, 1]$, instead of being binary. The term $\lambda^j$ refers to an update step. We assume that the master algorithm is applied to a pool of $n$ algorithms, letting $x^j_i$ denote the prediction of the $i$th algorithm of the pool in trial $j$. Let $\lambda^j$ denote the prediction of the
master algorithm in trial \(j\), \(\rho^i\) denote the result of trial \(j\) and \(w^j_1, \ldots, w^j_n\) denote the weights at the beginning of trial \(j\). Consequently, \(w^{j+1}_1, \ldots, w^{j+1}_n\) denote the weights after the trial \(j\). All initial weights \(w^1_i\) are positive. Let

\[ s^j = \sum_{i=1}^n w^j_i. \]

**Prediction:** The prediction of the master algorithm is

\[ \lambda^j = \sum_{i=1}^n w^j_i x^j_i. \]

**Update:** For each algorithm in the pool, the weight \(w^j_i\) is multiplied by a factor \(\theta\) that depends on \(\beta, x^j_i\), and \(\rho^i\).

\[ w^{j+1}_i = \theta w^j_i \]

where \(\theta\) can be any factor that satisfies

\[ \beta|x^j_i - \rho^i| \leq \theta \leq 1 - (1 - \beta)|x^j_i - \rho^i| \]

and \(0 < \beta < 1\).

### 3.3 Quality of Information

We adapt the algorithm to predict the reliability of a given sensor based on a set of reports from the sensor. Each fusion node maintains a weight \(w_i\) for each of the other sensors whose reports need to be fused. This weight estimates how reliable the given sensor is. However, applying the classical WMA for sensor systems presents a technical challenge, because sensor reports are not scalars, but rather belief functions. Therefore, our approach extends WMA to accommodate belief functions. Basically, our approach computes the difference between a prediction of fused sensor reports and each sensor report using the notion of conflict in Dempster-Shafer theory and accordingly updates the weight for each sensor.

Suppose sensor \(s\) wishes to evaluate the confidence of fused sensor reports from sensors \(\{s_1, \ldots, s_L\}\). The assumption here is that we should trust the majority of sensor outputs. Let \(s\) assign a weight \(w_i\) to sensor \(s_i\). The weight \(w_i\) is initialized to 1 if \(s\) gets the data from \(s_i\) for the first time.

**Definition 6** Given a sensor \(s_i\)'s probability mass function \(m_i\), its corresponding reliability factor (or weight) \(w_i\), the updated belief function can be defined as

\[ m'_i(A) = w_i m_i(A)/\sum_{i=1}^L w_i \]

\[ m'_i(\Theta) = w_i m_i(\Theta)/\sum_{i=1}^L w_i + 1 - w_i/\sum_{i=1}^L w_i \]

where \(A \subset \Theta\) and \(A \neq \Theta\).

The weight scheme can effectively mitigate the conflict between evidences, as two sensors will not be trusted equally anymore. Note that we use \(w_i/\sum_{i=1}^L w_i\) instead of \(w_i\) to adjust the sensor data, where \(0 \leq w_i \leq 1\). ^3

Similarly, we can give the prediction and updating rules for the reliability of each sensor during the process of decision fusion.

**Prediction:** The total belief of sensor \(s\) for target \(T_K\) is

\[ m' = m'_1 \oplus \ldots \oplus m'_L \]

where \(m'\) is the weighted average of belief functions \(m'_1, \ldots, m'_L\), and \(m'_i\) is the updated belief function, \(1 \leq i \leq L\).

Note that, when we consider the quality of sensor data in the fusion process, the confidence for target \(T_K\) needs to be revised as

\[ C(T_K) = \max_{v_i \in V} m'(\{v_i\}) + m'(\Theta). \]

**Update:** The weight of sensor \(s_i\) will be updated as follows.

\[ w'_i = \theta w_i \]

In weighted majority algorithm, \(\theta\) can be any value chosen between \(\beta \rho\) and \(1 - (1 - \beta) \rho\). For simplicity, we choose the upper bound as the value of \(\theta\).

\[ \theta = 1 - (1 - \beta) \phi. \]

Here \(\phi\) is the conflict between the prediction of fused sensor data and each sensor report in Dempster-Shafer theory. For a given sensor report from sensor \(s_i\), \(\phi\) is defined as

\[ \phi = \sum_{v_j \in V} m'(\{v_j\} - m_i(\Theta) - m_i(\{v_j\}) \]

where \(v_j\) is defined as \(\max_{v_i \in V} m'(\{v_i\})\) and \(m'\) is the basic probability assignment for the fused sensor data.

The above formula can be simplified as follows if \(\beta = 0.5\).

\[ \theta = 1 - \frac{\phi}{2} \]

Equation 10 indicates that the value of \(\theta\) relies on the conflict between the prediction \(m'\) and a sensor report. If the conflict \(\phi\) is large, \(\theta\) will be small. Correspondingly, the weight of the sensor is reduced to a low level.

Algorithm 1 describes the learning algorithm used by each sensor to adjust the weights of other sensors during the fusion process. We can find that the effect of the sensor is minimized in the fusion process when it returns unreliable sensor reports.

^3According to [24], the use of quality of information for decision fusion also solves the limitation of Dempster’s rule of combination for two totally conflicting events.
### Algorithm 1 Learning the quality of sensor data by sensor $s$

1. Initialize sensor $s_i$’s weight to 1 if $s_i$ returns the data to sensor $s$ for the first time
2. Let $\{s_1, \ldots, s_L\}$ be the sensors which return reports to sensor $s$ for target $T_K$
3. Sensor $s$ generates a prediction as specified in Equation 6
4. for each sensor $s_i$ do
5. Sensor $s$ computes the prediction from the report of $s_i$
6. Sensor $s$ updates the weight $w_i$ according to Equation 8
7. end for

### 4 Experiments

In this section, we first introduce a modeling and simulation environment, OTBSAF (OneSAF Testbed Baseline) \(^4\) [25]. And then we show a simple example of adaptive decision fusion rules and the learning algorithm for quality of information.

#### 4.1 OTBSAF System

OTBSAF models common military vehicles, aircraft, and sensors, and simulates uncertainty for entities’ individual and doctrinal behaviors in the battlefield. We extend OTBSAF and integrate it with our RETSINA multiagent system. One of our contributions to OTBSAF is to add three simulated mounted sensors, SARSim, EOSim, and GMTISim, to the simulation environment.

The SARSim simulates an automatic target recognition (ATR) system that receives its input from a synthetic aperture radar (SAR) that is operating in spotlight-mode. In spotlight-mode, a SAR scans an area of terrain, and the ATR will attempt to recognize any stationary object within the bounds of that scanned area. The output from the SARSim is a list of candidate target types (e.g., M1 tank, T80 Tank, etc.) with different confidence levels. While a real SAR/ATR system will report confidence levels for around three dozen entities, SARSim will report for a dozen entities. The GMTISim simulates a ground moving target indicator (GMTI) radar, which focuses a radar beam on one spot, and if it detects a moving target there with its ATR system, a motion tracker mechanism follows the movement of the target. While very similar in output and behavior to the SARSim, it is complementary, because it only recognizes entities that are moving, while the SARSim only recognizes entities that are stationary. The EOSim simulates an electro-optical sensor that detects targets at distances and in conditions in which they would be detectable in the ultraviolet, visible, and infrared light spectra.

Here we only consider the sensor data from SARSim. We assume there are multiple T80 tanks on the ground and a team of UAVs are scheduled to scan the area.

#### 4.2 Adaptive Decision Fusion

In the first example, we study the dynamics of the confidence for two T80 tanks for a series of sensor reports from sensors $\{s_1, s_2, \ldots, s_5\}$.

![Figure 5: An example of the dynamics of the fused sensor data confidence for two T80 tanks.](image)
Figure 5 shows the confidence for two targets #1 and #2 on the ground, where X-axis shows the number of sensor reports fused to get the confidence, Y-axis describes the confidence of fused reports. We can find that, for both targets, their confidence levels do not increase monotonically. For example, the confidence of target #1 drops from 0.49 to 0.28 after fusing the first two sensor reports. The reason is that the data from sensor $s_2$ misclassifies the target as an M1A1 tank. Moreover, the confidence levels for both targets reach a relatively high level. If we set the threshold $\omega$ as 0.6, the sensor $s$ which fuses these sensor reports can now recognize the type of target and initialize of a plan if necessary.

4.3 Learning the Quality of Sensor Data

In the second example, we illustrate the change of sensor reliability during the fusion process. Here we assume that sensor $s$ receives the reports for target #1 and #2 from the same set of sensors.

![Graph showing sensor reliability](image)

Figure 6: The sensor reliability after one and two rounds of sensor reports.

As we discussed in Section 3.3, we use a weight to denote the reliability of each sensor in the fusion process. The weight is initialized as one and updated according to the weighted majority algorithm. Figure 6 shows the sensor reliability after sensor $s$ receives the two rounds of sensor reports, one for target #1 and another for target #2 when $\beta = 0.5$.

From Figure 6 we can see that sensor $s_2$ returns low quality of sensor data. Consequently, sensor $s$ decreases its reliability (or weight $w_2$), so the effect of low quality data is minimized in the fusion process. Also, in the WMC version of weighted majority algorithm, the weight for each sensor $w_i$ may keep decreasing, since there is always a non-zero conflict between a single sensor report and the fused sensor data. In order to avoid this problem, we use $w_i/\sum w_i$ instead of $w_i$ in the fused process. Note that $w_i/\sum w_i$ may be greater than $w_i$ when $\sum w_i < 1$.

5 Conclusions

In this paper we present a novel approach to reasoning about uncertain and noisy sensor data in distributed temporal sensor systems. We incorporate data reliability into uncertain sensor data modeling within the framework of Dempster-Shafer theory, where each sensor learns reliability of other sensors using a novel application of the weighted majority algorithm.

One drawback of our approach is that it is based on a degree of consensus among various sensor sources. This may not work occasionally when the majority of sensors are unreliable and they provide conflicting evidence. In future work, we plan to explore the limitations of our approach. One way to address this drawback is to introduce some context information into the system, such as the expert knowledge in the sensor systems domain [26, 27]. We also would like to investigate the effectiveness of our approach with large amounts of sensor data from various sensors and its effects on high-level information fusion, such as force structure recognition and intent inference [28, 29].

Acknowledgments

The authors would like to thank Joseph Giampapa, Sean Owens, Robin Glinton, Young-Woo Seo, Charles Grindle, and Dr. Michael Lewis for their contribution to the system development and their helpful comments. The authors would also like to thank our partners at Northrop-Grumman for their assistance in developing some of the external sensor simulators. This research was supported by AFOSR under grants F49640-01-1-0542, and by AFRL/MNK grant No. F08630-03-1-0005.

References


