Reasoning Frameworks for Fusion of Imaging and Non-imaging Sensor Information

Pierre Valin[a1]

[a] Lockheed Martin Canada, 6111 Royalmount ave., Montréal, Canada

ABSTRACT

Reasoning over attributes or situations plays a big role both in military and commercial domains. The main focus here is reasoning frameworks for uncertain and incomplete information that can lead to a correct understanding of the military picture at all levels of fusion, leading to refinements in single object (Level 1), situation (Level 2) and implication (Level 3). The attributes, over which one has to reason at Level 1, can originate from imaging or non-imaging sensors and be kinematical, geometrical or relate more directly to the ultimate obtention of a unique identification (e.g. from Identification Friend or Foe (IFF), Electronic Support Measures (ESM), or imagery classifiers). The concept of a unique identification (ID) itself contains an ambiguity related to the desired taxonomy, over which reasoning occurs, and this choice is discussed with respect to the operator's need for a decision aid tool. An example set of databases is discussed that contains platform characteristics, emitter-to-platform mapping, geopolitical allegiance information, etc. Physical attributes often require fuzzification for proper treatment by evidential reasoning. The following four reasoning frameworks are introduced and examples are given for a) fuzzy logic, b) Neural Networks (NNs), c) Bayesian approach (with a priori information), and d) Dempster-Shafer (DS) approach with several variants.

1. INTRODUCTION

According to the updated 1999 Joint Directors of Laboratories (JDL) classification of Data Fusion (DF) levels, one can expect reasoning at correspondingly different levels:

- Level 0: *sub-object assessment* should require only pre-processing possibly selected from a priori knowledge of possible acquisition problems or difficult situations
- Level 1: *single object refinement* should involve evidential reasoning over single object kinematics and attributes, towards the goal of obtaining the best platform ID or at least some level of the taxonomy tree
- Level 2: *situation refinement*, a.k.a. Situation and Threat Assessment (STA) should involve reasoning over groups of objects and proceed by higher inference rules involving doctrinal and contextual information
- Level 3: i*mplication refinement*, should involve reasoning over plan alternatives to suggest plan decisions
- Level 4*: process refinement* should involve reasoning over ownship and environmental conditions in order to perform better sensor management and thus close the Observe, Orient, Decide, Act (OODA) loop.

Here one concentrates mostly on single object evidential reasoning [1] (mostly algorithmic, fuzzy...), and touches on reasoning over groups of objects (mostly knowledge-based or involving Artificial Intelligence (AI) reasoning).

2. ATTRIBUTES CHARACTERISTIC OF PLATFORMS AT LEVELS 1 AND 2 OF FUSION

As a preamble, one should be aware that the format in which the databases should be encoded is an active topic of research, with experts still debating whether relational databases or Object-Oriented (OO) databases are preferable. Compromises often have to be made and the use of Commercial-Over-The-Shelf (COTS) software (SW) may in the future make as many inroads as COTS hardware (HW) is currently doing throughout the world's military establishment. This section only describes the content of these databases, not their optimal architecture nor how complex their interrelations are woven.

---

[1] Further author information: (Send correspondence to the following e-mail: pierre.valin@lmco.com)

2.1. The Platform Database and Related Namelists

For ID estimation to be properly achieved in Level 1 DF, all possible attributes that can be measured by all sensors must be listed in the Platform Database (PDB). They can be split into three groups [2,3]:

1. *Kinematic attributes*: which can be estimated through tracking in the positional estimation function of DF, and through reports from IFF and datalink. The maximum acceleration, the maximum platform speed, the minimum platform speed and the cruising speed either serve as bounds to discriminate between possible air target IDs or suggest the most plausible IDs. The maximum altitude that a platform may reach can serve as a bound for altitude reported by the IFF or deduced from a tracker in 3-D from sensor reports in 2-D [4].
2. *Geometrical attributes*: which can be estimated by algorithms which post-process imaging information from sensors such as Forward Looking Infra-Red (FLIR), or Electro-Optics (EO) and Synthetic Aperture Radar (SAR). Classifiers that perform such post-processing can be thought of as Image Support Modules (ISM) performing much the same functionality as the ESM does for the analysis of electromagnetic signals. These ISMs need the three geometrical dimensions of height, width and length (for FLIR and EO), and also Radar Cross Section (RCS) of the platform seen from the front, the side and the top (for the SAR and radiometric radar). In addition, the distribution of relevant features may be needed for classifiers, but may be considered part of the algorithms that generate plausible IDs.
3. *Identification attributes*: which can be directly given by the ESM, or as outputs of the FLIR and SAR ISM. The ESM requires an exhaustive list of all the emitters that are carried by the platform. However the ISMs are usually designed to not only to provide the best single ID possible but also to estimate confidence in higher levels of an appropriate taxonomy tree (discussed in the next section). This taxonomy tree is usually derived from some standard, either STANAG 4420 or MIL-STD 2525 (A or B), or can even be customized depending on the application (naval, airborne or helicopter platforms). This is indeed the case for the applications studied by Lockheed Martin Canada (LM Canada), namely the Canadian Patrol Frigate (CPF), the maritime surveillance aircraft CP-140 Aurora (a P3-C with S-3B avionics), and the LAMPS helicopter.

Some sensors measure attributes quite directly. For example the ESM will provide an emitter list with some confidence level about the accuracy of the list that reflects the confidence in its electromagnetic spectral fit. However an IFF response can lead to an identification of a friendly or commercial target but the lack of a response does not necessarily imply that the interrogated platform is hostile. One has to distribute the lack of a response between at least two declarations: the most probable foe declaration and a less probable friendly or neutral declaration that allows for an IFF equipment that is not working or absent.

Some complications arise when dealing with kinematic parameters reported occasionally by the tracker in *positional estimation*. First, each physical quantity has a different dimension (speed, acceleration) and an accurate determination is not necessarily needed for fusion. Indeed it is convenient to bin the attribute “speed” into fuzzy classes like “very fast”, “fast”, “average”, “slow” and “very slow” (separately for air and surface targets). The same can be done for length in bins of 40-meter width, for example. Membership in each class is a measure of how well the measured value fits into the descriptor as described below. Thus fuzzyfication is necessary even though fuzzy logic may not be used as a reasoning framework. Similarly defuzzification may be used to present results to the operator through the Human Computer Interface (HCI).

Other complications arise in DF with respect to correlating tracker speed values with the speed attribute in the PDB. Indeed, speed reports must be fused only if they involve a significant change from past historical behaviour in that track. The reason is two-fold. First, no single sensor must attempt to repeatedly fuse identical ID declarations, otherwise the hypothesis that sensor reports are statistically independent is violated. Second the benefits of the fusion of multiple sensors is lost when one sensor dominates the reports. Furthermore, a measured value of speed only indicates that the target is capable of that speed, not that it corresponds to either the maximum or minimum speeds listed in the PDB. It is a reasonable working hypothesis to fuzzify the value reported by the tracker into adjacent “bins” to account for the target being at, say, only 80% of optimal speed (a “very fast” target can occasionally travel “fast”), or travelling with a strong tailwind (a “fast” target can appear as “very fast”). Finally the concept of binning can be generalized to continuous membership functions of a fuzzy set.

Although the PDB is the most important of the databases and is immutable, there are many supporting databases which are necessary and flexible, such as the Geo-Political Namelist (GPL), which contains acronyms of the countries that own platforms in the PDB. In the PDB, the acronym is used by the attribute fusion function to link the PDB platform with the country allegiance indicated in the GPL: friendly, foe or neutral (depending on the mission). In case language communications can be intercepted, the languages spoken in different countries are also listed. The Emitter Namelist (ENL) offers a correspondence between the index number of each emitter and its name (actual or given by NATO), various characteristics (Pulse Repetition Frequency (PRF), frequency, detection range, etc.) and can offer (optionally) a cross-listing of platforms on which each emitter can be found.

2.2. Choice Of A Taxonomy Tree

The breadth and width of a taxonomy tree can easily grow out of proportion. This is where the decision on the database architecture plays a major role. Figure 1 illustrates this fact for a taxonomy tree of depth 2 for air and surface targets (not counting the final ID level) [5].

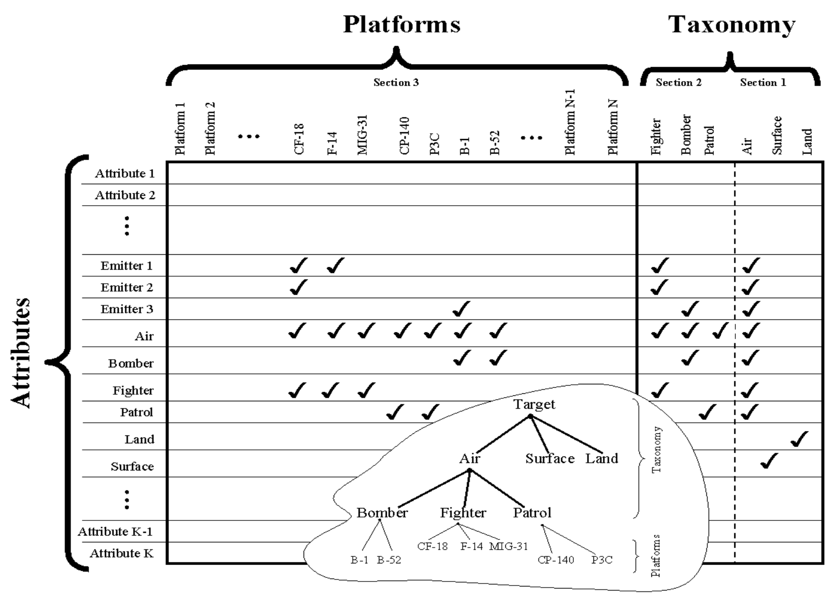

Figure 1. Example PDB with Platforms and the Taxonomy Tree

In practice, there are many more levels in a taxonomy tree. As a pedagogical example, the first level of detail should choose between AIR, SEA SURFACE, SUB-SURFACE and LAND (or GROUND). The next finer levels (e.g., class and sub-class) could offer the following branches, shown in Table 1 for ground and sea-surface nodes.

Table . Taxonomy Example for Ground and Sea Surface Targets

Level 2	**Level 3 - GROUND**	**Level 2**	**Level 3 – SEA SURFACE**
Vehicle	Armoured	Combatant	**Line**
	Engineer vehicle		Amphibious warfare ship
	Train locomotive		Mine warfare vessel
	Civilian vehicle		Patrol
Weapons	Missile launchers		Hovercraft
	Single rocket launcher		Station
	Multiple rocket launcher		Navy group
	Antitank rocket launcher	Non-combatant	Underway replenishment
	Other		Fleet support

Level 2	Level 3 - GROUND	Level 2	Level 3 – SEA SURFACE
Sensor	Radar		Intelligence
	Emplaced		Service & support harbour
Special	Laser		Hospital ship
	NBC equipment		Hovercraft
	Flame thrower		Station
	Land mines	Non-military	**Merchant**
Installation	material facility		Fishing
	ship construction		Leisure craft
	government leadership		Law enforcement vessel
	airport		Hovercraft
	other		Emergency or Hazard

The branch terminations listed above serve as nodes for finer branches, e.g., specific classes, leading eventually to the single platform ID. Noteworthy examples (shown in **bold** in Table 1) are **line** ships separating into Frigate/Corvette, Destroyer, Cruiser, Battleship, Aircraft Carrier, and **merchants** separating into Cargo, Roll-on/Roll-off, Oiler/Tanker, Ferry, Passenger, Tug, etc.

2.3. Database for Higher Levels of Fusion

It seems natural that Level 2 fusion accesses a different database than the PDB relevant for Level 1. The STA/RM-DB should contain all the platform parameters relevant for STA as well as RM, i.e., since missiles (number and detailed properties) on enemy ships are relevant for STA, while the same information on possible own-platforms is relevant for RM. In a Cooperative Operating Picture (COP) context, the lethality of enemy platforms (the red force) is important for STA, and the lethality of cooperating Participating Units (PUs) is relevant for RM within the blue force. A non-exhaustive list contains for SEA SURFACE elements pertaining to:

1. Platform and Mission: Displacement (in tons under full load), number of hulls (sea-worthy examples of the ID), list of hull numbers, list of hull names, platform type, amplification, and role, crew (for full operation), operational or standby flags.
2. Armament: Number of Surface-to-Surface Missiles (SSMs), Surface-to-Air Missiles (SAMs), Close-In Weapon Systems (CIWSs), guns, torpedo tubes, troop complement (number of special force people for assault or landing), lethality, and flags for Electronic Counter-Measures (ECM), acoustic CM, infra-red CM (e.g., chaff) equipment.
3. Sensors: Number and list of radars, sonars (e.g., Hull-Mounted Sonar (HMS), towed-array, sonobuoy, tethered sonar), and imaging sensors (e.g., EO, FLIR, IRST, SAR).
4. Air Platforms on Deck: Number of helicopters, aircraft, with PDB index list.

and similarly for AIR platforms with the same decomposition (item 4 above cannot exist):

1. Platform type, amplification role, number of wings (examples in flying conditions), list of wing number, range, crew (number of people for full operation), operational or standby flags.
2. Number of Air-to-Surface Missiles (ASMs), Air-to-Air Missiles (AAMs), CIWSs, conventional bombs, troop (number of special force for assault or parachuting), lethality, and flags for ECM and infra-red CM equipment.
3. Number and list of radars, number and of imaging sensors (e.g., EO, FLIR, SAR), guidance.

3. FUZZY LOGIC REASONING

Fuzzy logic deals with approximate modes of reasoning. In standard logic, a proposition is either true or false. In fuzzy logic a proposition has a parameter value, called a “membership value”, ranging from 0 (completely false) to 1 (completely true). Zadeh’s fuzzy logic is a well-defined formalism that describes the fuzzy propositions and combination rules to create syllogisms and inferencing for using fuzzy probability. Fuzzy logic application to the ID estimation problem is not as well documented in literature as are the Bayesian or DS approaches.

Information from the ESM sensor can be considered fuzzy. The ESM measures few parameters and the results of the measurement process are compared with a database of emitter characteristics. The measured parameters can be corrupted by counter-measures, making the comparison uncertain. Also, information in this database can be considered fuzzy, to a certain extent, due to inherent imprecision. Bayesian or DS approaches presuppose that input data follow strict logic rules of statistics or probability. If sensor declaration “noise” is a little different from the measurement noise or if the sensor declarations are not completely independent, the Bayesian or DS approaches produce results that are quantitatively inexact but qualitatively correct. In other words, they are neither false nor true as per Boolean logic. This is exactly the character of a fuzzy quantity. In this section, despite the absence of literature about the attribute combination for ID estimation, a simple fuzzy logic algorithm that combines the ESM sensor evidence is developed as a pedagogical example. The algorithm below is an *ad hoc* algorithm that imitates the phenomenology coming out of a fuzzy logic estimator. The example does not cover all particularities of fuzzy logic rules since the information comes from a single ESM sensor and is always of the same form. Nevertheless, the example permits appreciation of the fuzzification of the input and the defuzzification of the output processes with their interpretation.

A typical fuzzy logic algorithm (used here as an estimator) is composed of three fundamental processes: the input fuzzification process, which combines the fuzzy variables, the output defuzzification process together with the knowledge base source that contains the inference rules, and the logic combination rules. The input data from the sensor are first interpreted and put in a fuzzy variable form in order to perform normalization and interpretation of the input data. When the data are all numerical, all of the same type and from a single source, the interpretation task is relatively easy. The combination process performs the arithmetic or the algebra that is necessary to combine the input fuzzy variables, under the recommendation of the inference rules or logic combination rules. The defuzzification process performs the opposite task of the first process: it transforms the fuzzy statements into the appropriate information type

3.1. Example of Fuzzification Rules

To be able to apply this method, the ESM sensor is required to output a list of candidate emitter identities with a Confidence Level (CL) associated with each emitter. This CL should be a numerical representation of the quality of the fit between the emitter characteristics and the measured parameters. For each pre-determined CL range a fuzzy statement is associated. As ranges descend, the fuzzy statements are less and less categoric about the possibility of the ESM declaration. The third column shows the numerical weight, which is given to the fuzzy statement and used by the combination process.

Table 2 . Fuzzification Rules for an ESM Sensor

ESM Computed CL	Associated Fuzzy Statement	Numerical Weight
$1 \geq Conf > 0.99$	Extremely Possible	3
$0.99 \geq Conf > 0.95$	Highly Possible	2
$0.95 \geq Conf > 0.70$	Very Possible	1
$0.70 \geq Conf > 0.50$	Possible	0
$0.50 \geq Conf > 0.25$	Moderately Possible	-1
$0.25 \geq Conf > 0.02$	Slightly Possible	-2
$0.02 \geq Conf > 0.005$	Almost Impossible	-3
$0.005 \geq Conf \geq 0$	Impossible	-4

As already mentioned in Paragraph 2.1, the fuzzification process can be generalized in terms of membership functions, where the crude binning of the example above is replaced by overlapping functions which sum up to one, usually of triangular or trapezoidal shape. An example of a 5 term decomposition into triangular membership functions is shown in Figure 2 below for the physical attribute of air speed (VS = Very Slow, S = Slow, M = Medium, F = Fast, VF = Very Fast). The overlapping regions measure the fact that experts need not agree on a precise definition of linguistic terms for speed.

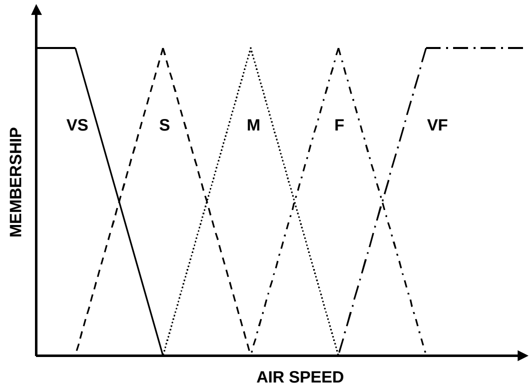

Figure 2. Fuzzy Membership Functions for Speed

3.2. Example of Combination Process

The combination process is performed in two steps. First, the input possibility is combined with the previous combined possibility and second, the last five combined possibilities are averaged to get the track level output possibility. For each emitter E_i, the track level possibility at any time t_n is computed by

$$Copos(E_i,t_l) = Copos(E_i,t_{l-1}) \otimes InPos(E_i,t_l)$$

$$Pos(E_i,t_n) = \frac{1}{h}\sum_{l=n-h+1}^{n} CoPos(E_i,t_l) \qquad h = Smaller(5,n)$$

$$Pos(E_i,t_1) = CoPos(E_i,t_1) = InPos(E_i,t_1)$$

where the last equation gives the initialization condition. *Pos, CoPos* and *InPos* are the output track level possibility, the combined possibility and the input possibility (contact level), respectively. The index h, chosen here, as an example, to be the lesser of n and 5, restricts the average to the last five values of combined. The ⊗ symbolizes the application of the combination rule, which is represented by the following inference matrix. The matrix in Table 3 provides the result (using the numerical weight) of the combination of the new input information with the previous one. The rationale concerning the choice of the elements is essentially system performance dependent and based on the experience of the designers with the problem of ID estimation. As the symmetry among the numbers in Table 3 shows, the combination rules favour the new input data more than the previously combined data. This is counterbalanced by the averaging process over the last five samples.

Table 3. Inference Rules Combining the Last CoPos with the Input Possibility

	Last CoPos							
InPos	**3**	**2**	**1**	**0**	**-1**	**-2**	**-3**	**-4**
3	3	3	3	3	2	1	0	-1
2	3	3	3	2	1	0	-1	-2
1	3	3	2	1	0	-1	-2	-3
0	2	2	1	0	-1	-2	-3	-4
-1	1	1	0	-1	-2	-3	-3	-4
-2	0	0	-1	-2	-3	-3	-3	-4
-3	-1	-1	-2	-3	-3	-3	-3	-4
-4	-2	-3	-3	-4	-4	-4	-4	-4

Finally defuzzification rules can proceed by a table similar to Table 2 or by slight variations of it.

4. NEURAL NETWORKS (NN)

NNs are particularly useful when one has a large volume of data to reason over, such as a large collection of image features that are to be used for the purposes of classification. This is the case of FLIR and SAR imagery. A typical NN application starts out by extracting features from typical imagery, e.g., invariant moments and auto-regressive model parameters for FLIR ship images. Depending on the size of the training set and the independence of the input attributes, a number of hidden layers containing a selected number of neurons with intricate interconnections is chosen and the NN is trained on a fraction of the available imagery. The remaining imagery is kept for the validation and test sets.

A practical example of such a FLIR classifier is shown in Figure 3 where 11 inputs are used to identify 8 types of ships obeying a taxonomy tree similar to the one discussed previously, namely: destroyer, frigate, cruiser, destroyer with guided missiles, landing assault tanker, auxiliary oil replenishment, civilian freighter, and cargo/container. The taxonomy tree regroups the first four as “line” ships, the last two as “non-naval”, while “line” and the landing assault tanker form a “combatant” group (borrowing from STANAG 4420 terminology). The auxiliary oil replenishment ship is the lone example of the non-combatant type. There were 2545 images available (close to the rule of thumb 30*11*8).

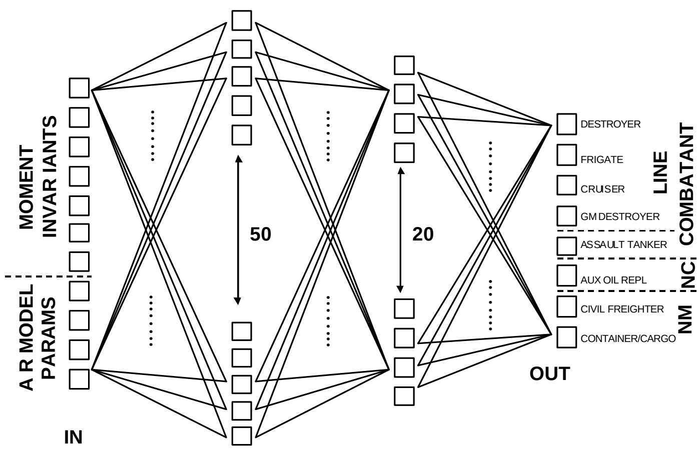

Figure 3. FLIR Classifier with Two Hidden Layers, 11 Inputs and 8 Outputs

An example of the training, validation and test steps of NN classifiers is shown below for the SAR category classifier for ships into line or merchant categories. One first decomposed the range profile of ships into 9 bins, then selected 32,211 profile vectors which, according to classified knowledge based rules, correspond to the following ship distribution: 16,259 line combatants, 5,922 merchant ships, 10,030 others.

The 9-bin profile vectors were normalized such that the sum was one and no given cell exceeded 0.4. Following the usual claims that a single hidden layer NN can classify any problem as long as enough neurons are taken, one has chosen to optimize a single layer NN by varying the number of hidden neurons. The next step was to vary the training set size, given a fixed (but randomly chosen) validation set of 1000 profile vectors. The values used for the training set size were thus varied through 100, 200, 500, 1000, 2000, 5000, 10000. Figure 4 shows a few training set curves. The top curve shows the generalization error and the lower curves the learning error. The upper left and right sub-figures, which correspond to 100 training examples, show a marked difference between the two curves, which is an indication that the system has memorized the examples. One can conclude that 100 examples for training is too little. However, the lower left and right sub-figures, which correspond to many more examples (5000), have very similarly shaped curves and the generalization curve (upper curve) flattens out rather than increases, which is the desired behaviour.

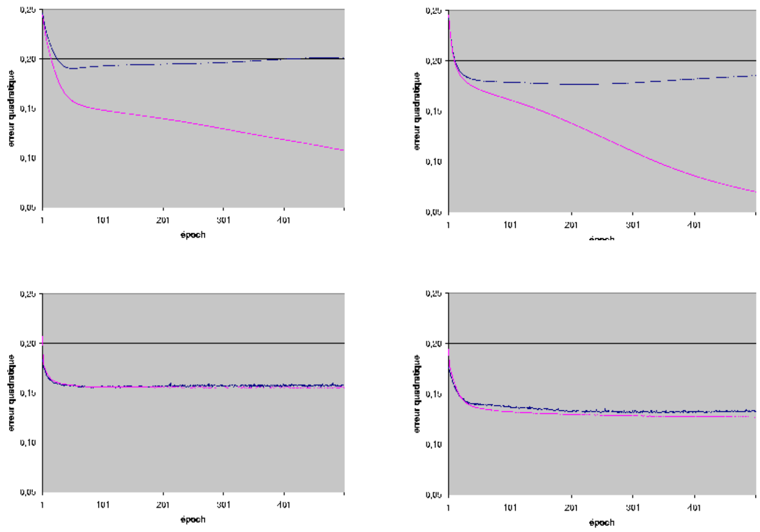

Figure 4. Training Set Curves for 100 or 5000 Examples and 6 or 19 Neurons

5. BAYESIAN REASONING

Bayesian reasoning is at the foundation of many tracking and evidential reasoning approaches. One concentrates here on its application as an alternative classifier to the NN approach outlined in the previous section, for ship type (rather than category), extracted from SAR imagery [6]. If the confidence level on a "line" declaration is high enough (lets say > 50%) from the NN, then an estimate of the line ship type should be initiated. This is performed using a Bayes classifier based on the frigates, destroyers, cruisers, and aircraft carriers (porte-avions in French) length distributions (again borrowing from STANAG 4420 terminology). Ship lengths for this statistical analysis have been obtained by browsing Jane's Fighting Ships and their probability density distributions approximated by the curves shown in Figure 5 with the types listed above from left to right

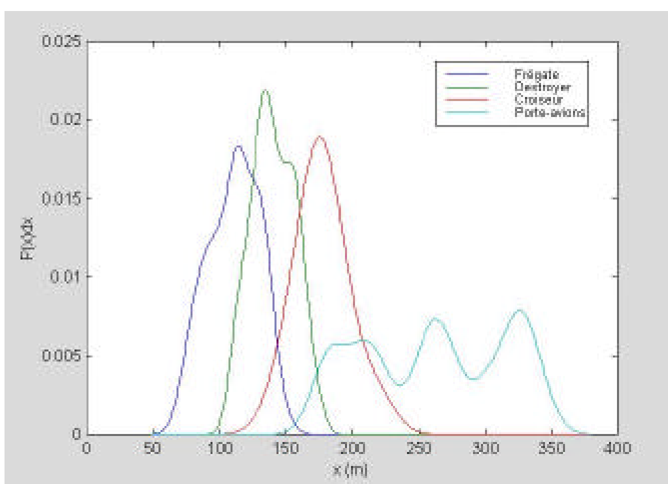

Figure 5. Probability Density Distribution Model for Line Ship Length

Given that a ship length range has been evaluated from the ship end points in the imagery, one calculates the mean *a posteriori* type probability $P_{avg}(t|s)$ that a ship s belongs to type t by averaging the standard Bayes rule over the entire ship length range,

$$P_{avg}(t|s) = \underset{length\ range}{Avg}\left(\frac{p(s|t)P(t)}{p(s)}\right) \qquad\qquad p(s) = \sum_i p(s|t_i)P(t_i)$$

where $p(s|t)$ and $P(t)$ are the probability density distribution and the *a priori* probability of type t, respectively. Mean *a posteriori* type probabilities are re-normalized, in order that their sum is unity. Naturally the *a priori* probability $P(t)$ depends on the context and could be set by the radar operator prior to the mission.

Given the four following a priori distributions of Table 4

Table 4. A Priori Probabilities for Four Types of Line Combatants

Figure	Frigate	Destroyer	Cruiser	Aircraft Carrier
a	0.25	0.25	0.25	0.25
b	0.10	0.10	0.70	0.10
c	0.10	0.70	0.10	0.10
d	0.15	0.70	0.15	0.0

one obtains the following probabilities shown in Figure 6

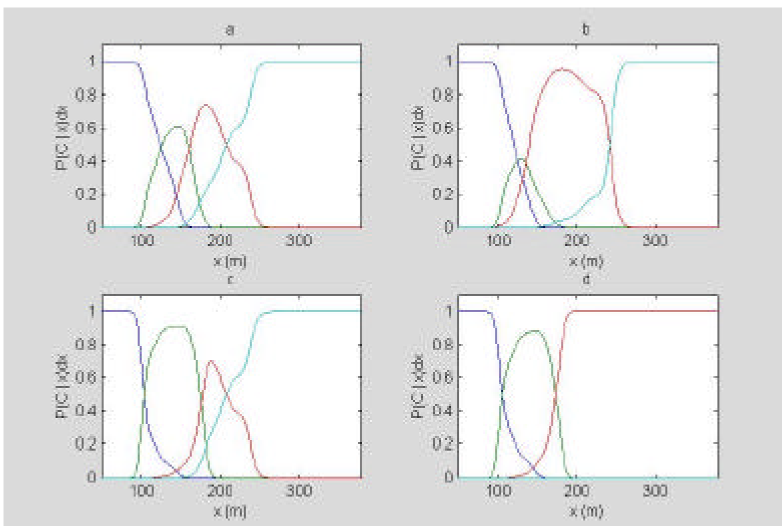

Figure 6. Probability Distributions of Line Combatants for Different A Priori Probabilities

The performance of this Bayesian classifier on the actual database of 145 ships used, given an equal 0.25 a priori probability of occurrence of any particular type, is shown in the confusion matrix of Table 5.

Table 5. Bayesian Line Type Performance for Equal A Priori Probabilities

	Frigate	Destroyer	Cruiser	Aircraft Carrier
Frigate	37	13	-	-
Destroyer	9	34	4	-
Cruiser	-	3	13	2
Aircraft Carrier	-	-	10	10

6. COMPLEX PROBLEMS MAY REQUIRE HYBRID METHODS

The above discussions about NN and Bayes techniques show that they can compete in the designer's mind, but it should be reminded that they can also be complementary. Indeed both have different required assumptions that must hold, preferred regions of applicability, and performance measures which vary both in classification accuracy and in computing time, stability and complexity. Hence a hybrid design, such as the one presented in Figure 7 can exploit the strengths of both approaches while avoiding any shortcomings [7].

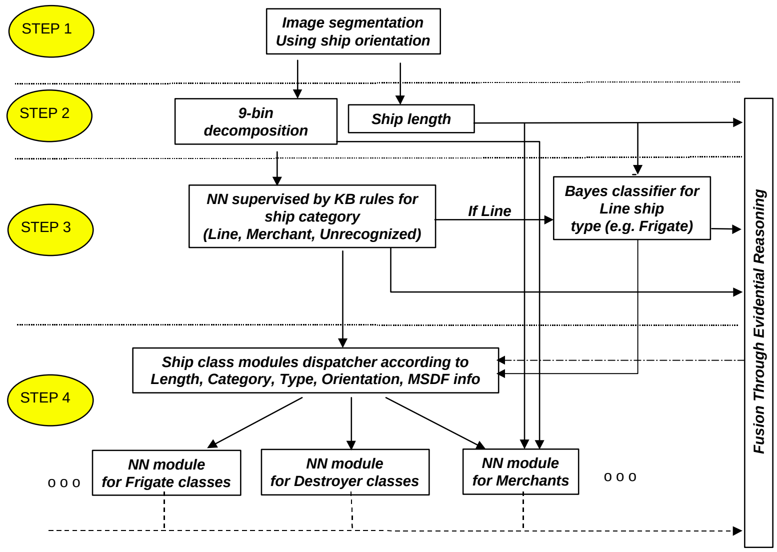

Figure 7. Hybrid SAR ISM Classifier

In Step 1, target segmentation from the ocean clutter is achieved by noise removal and merging of small regions. The resulting segmented image is made binary and sent to the ship centerline detection algorithm. Ship end-points can be obtained by estimating the ship centerline from the maximum peak of the Hough transform of the segmented image. This technique is more robust than a least-square-fit through the target or principal axes detection, because a large amount of cross-range scatterer spreading in some images (especially ISAR) tend to bias the centerline estimation. Ship length in Step 2 is obtained by identifying the target's end-points, using an empirical formula, and then sent to the Bayes length classifier. Simultaneously, the range profile is decomposed into 9 bins (by counting the number of pixels within that bin) and sent to the NN supervised by the classified knowledge based rules.

7. DEMPSTER-SHAFER EVIDENTIAL REASONING

The mathematical formalism of the DS evidential theory requires that propositions pertain to the set of all possible propositions that can be output. More precisely, using the language of the set theory, a proposition is a set where each irreducible element (elementary proposition) pertains to a complete set named the "frame of discernment" usually noted Θ. If "n" is the cardinality of Θ (number of elements) there will be 2^n possible propositions that can be raised from Θ. The set of all possible propositions is usually noted as 2^Θ. The Basic Probability Assignment (BPA), also known as mass, is a value associated by a sensor with the proposition that indicates the level of confidence or certainty given by the sensor.

The mass is a quantity output by a sensor which infers information only about the proposition "A" it has deduced and nothing at all about the propositions which are subsets of A. A mass close to one is a strong indication that this proposition is probably true. However, a mass close to zero does not necessarily mean the opposite, since this depends on the mass given to other alternative propositions, including the ignorance. In DS all the information about a proposition is obtained from a pair of quantities which are computed from the masses of various "related" propositions. The DS approach defines the concept of an evidential interval, denoted *[Bel, Pls]*, where the lower bound, the belief *Bel*, and the upper bound, the plausibility *Pls*, are obtained from

$$Bel(A) = \sum_{i=1}^{2^{|A|}} m(A_i) \qquad where\ \ all\ \ A_i \subset A \qquad\qquad Pls(A) = 1 - Bel(\neg A)$$

The belief in a proposition "A" represents the minimal commitment, which can be extracted by the masses from various sensor-declared propositions B_j. All B_j that are subsets of A contribute to that minimal commitment. The plausibility of a proposition "A" represents the maximal commitment from the sensor declarations. All B_j that have at least an element of Θ in common with A contribute to the plausibility. Consequently the Bayesian Probability (*Prob*) of the proposition A satisfies

$$Bel(A) \leq Prob(A) \leq Pls(A)$$

Finally, there are other evidential intervals which can be defined in a way analogous to *[Bel, Pls],* such as the Expected Utility Interval (EUI).

7.1. Combination Rules

Conventional literature allows for the traditional combination rules where conflict can exist between the latest sensor declaration about a target and the existing cumulative knowledge about that platform in the track database [2, 6, 8, 12, 14]. More recent literature [1] allows for not declaring a conflict at a finer level of the taxonomy tree but rather for throwing back the "conflict" to the first coarser level of taxonomy where conflict does not exist. These combination rules are referred to as the "orthogonal sum" and the "hierarchical orthogonal sum", respectively, and are discussed below.

Suppose A_i and B_j are members of two lists of sensor propositions which are statistically independent, i.e., their mass values are not correlated or computed from the a priori knowledge of each other. The numbers of members in each list are α and β respectively. As with Bayes formula, which combines independent probability measurements, DS evidential theory has a way to combine independent mass measurements to get new mass data on the output propositions and also some propositions obtained from combined propositions. The frame of discernment Θ should be the same for both lists of declarations. If the masses associated with propositions A_i and B_j are independent, the new belief of the intercept of both propositions is obtained from DS's rule of combination, a.k.a. the orthogonal sum:

$$m(C_k) = \sum_{i,j \text{ where }} \sum_{A_i \cap B_j = C_k} \frac{m(A_i)m(B_j)}{1 - K}$$

$$K = \sum_{i,j \text{ where }} \sum_{A_i \cap B_j = \varnothing} m(A_i)m(B_j)$$

where K is called the conflict. The applications of these two formulas are not straightforward when the amounts α and β are large. When the number of reported propositions is high, the combination rule has a tendency to increase the number of propositions by creating new ones. The problem is an NP-hard one and a truncation scheme has to be devised. In other words, despite the fact that the orthogonal sum takes conflict into account in a mathematically correct way, this may not be sufficient for the stable operation of DF in a decision aid role.

In the case of the existence of a taxonomy tree, one can do better than the orthogonal sum's resolution of conflict. In order to apply the DS evidential combination method efficiently, information has to be mapped into proposition bitstreams. For example, if the report Air is received from a sensor, and the platforms or taxonomy elements #1, 2 and 3 are members of set "*Air*", then the bits #1, 2, and 3 of the proposition will have the value "1". All other elements that do not support the input information shall have their corresponding bit set to *off* (i.e., "0"). Figure 8 shows an example where an input attribute *Air* is supported by 21 elements.

0	1	1	0	1	1	1	• • •	1	0	• • •	0	1	0
bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7		bit 22	bit 23		bit 1022	bit 1023	bit 1024

Figure 8. Proposition from an Input Attribute for 1024 Platforms and Taxonomy Elements

The peculiarity of the Hierarchical Orthogonal Sum (HOS) is that one implements bitstreams based not only on the platforms but also on the whole taxonomy tree [5]. There is the implementation of a hierarchical structure with AND/OR addition rules in the orthogonal sum instead of only an AND rule as in the standard DS method. If one looks at the taxonomy tree in the right-hand side of Figure 1 an incoming report "*Air*" would not only fill the bits corresponding to all air platforms but would also fill the bits corresponding to Fighter and Bomber (they have a member relationship with the report "Air") and the bit corresponding to "*Air*". This is useful because it allows a better treatment of conflicting reports. In the standard DS method, addition of information is done through the use of a logical AND applied only on the platform elements.

Algorithmically, each level in the taxonomy tree has a section in the bitstream: section[I] with I=1,2,3. Section[1] corresponds to Level 1, containing the section Air, Land, Surface; Section[2] to Level 2, containing the portion Fighter, Bomber; and the last level, Section[3], corresponds to the platform elements portion as in Figure 1. The HOS algorithm goes as indicated in the following text box. In this box the operator "==" refers to the standard equality comparison operator, $\cup$ and $\cap$ refer to the logical OR and AND operations applied on bitstream, and the $\oplus$ symbol stands for the concatenation procedure applied on a section of bitstream.

Given two propositions described by the bitstreams A and B:

Add: A.section[3]$\cap$B.section[3]==$\emptyset$
- **No:** C=A.section[1,2,3]$\cap$B.section[1,2,3]
- **Yes:** A.section[2]$\cap$B.section[2]==$\emptyset$
  - **No:** C={(A.section[3]$\cup$B.section[3])}$\oplus${A.section[1,2]$\cap$B.section[1,2]}
  - **Yes:** A.section[1]$\cap$B.section[1]==$\emptyset$
    - **No:** C={A.section[2,3]$\cup$B.section[2,3]$\oplus${A.section[1]$\cap$B.section[1]}
    - **Yes:** C=Ignorance.

This algorithm allows us to move up the taxonomy tree to the last non-conflicting information level ("Air" in our case) and provides a better treatment of conflicts. At the fusion step we use the whole taxonomy. When calculating a score through the association step, one however compares only based on the platform bits. Thus in the last case, the conflict between the two reports would be reported fully. This is necessary because prior to fusing a new proposition we need to gate the incoming information with the one available. This also allows us to better report information when there is conflicting information: if in our previous example, the conflicting report had a very strong belief value, then with the standard DS method, the statement "Air" would have been lost until other reports appeared. With the HOS method, we kept the information available at all times.

7.2. Truncated Demspter-Shafer for real-time operation

The DS theory of evidence, because it assigns BPA to subsets of a PDB and not to its individual elements, can generate an exponential number of propositions, which is not very useful if used in real time applications. This is where the truncation algorithm comes in. The Truncated DS (TDS) scheme retains propositions according to the rules below (or equivalent ones) [8]:

1. All combined propositions with BPA > MAX_BPM are kept
2. All combined propositions with BPA < MIN_BPM are discarded
3. If the number of retained propositions in step 1 is smaller than MAX_NUM, it retains, by decreasing BPA, propositions of length 1
4. If the number of retained propositions in step 3 is smaller than MAX_NUM, it does the same things with propositions of length 2
5. Repeat a similar procedure for propositions of length 3
6. If the number of retained propositions is still smaller than MAX_NUM, it retains propositions by decreasing BPA regardless of length.

The propositions that are discarded are either returned to the ignorance (in most applications) or split between the ignorance and the remaining propositions (in some applications). Optimization and benchmarking [2] on complex realistic scenarios have catalogued the possible values of the three parameters MAX_BPM, MIN_BPM, and

MAX_NUM, as well as their interrelationship and their dependence on PDB size. A slight variant of this scheme is presently being implemented for the US Navy's LAMPS helicopter for real-time operation.

7. CONCLUSION

This lecture has provided an exhaustive list of platform attributes needed for evidential reasoning techniques that aim to provide an ID within a well-defined taxonomy tree. It also has presented the formalism and real-world examples for 4 well-known reasoning frameworks, mostly applicable to Level 1 and 2 fusion, a partial list being:

1. fuzzy logic, in particular its use in fuzzification (pre-processing) for other means of reasoning, but also with an ESM application utilizing fuzzy combination rules and defuzzification
2. NNs, particularly useful for large sets of data, such as imagery datasets, which are easily decomposable into training, validation and test sets, with FLIR and SAR examples detailed
3. Bayesian approach (with a priori information) with an application to classifiers where an attribute stands out as a discriminator (such as line ship length for a SAR classifier)
4. DS approach with its variants: the original orthogonal sum and its renormalization though conflict, followed by the HOS and its novel conflict resolution through the taxonomy tree, and finally generic considerations about truncating the exponentiation of proposition (NP-hard aspect of the problem).

In addition, examples were given, where all of the above methods form crucial parts of a larger versatile classifier.

ACKNOWLEDGMENTS

The author would like to thank all the LM Canada researchers listed in the references below, with whom I have worked over the last 8 years, and the graduate students whose thesis research was also instrumental. Further references to earlier works can be found in the references below.

REFERENCES

1. P. Valin, "Reasoning Frameworks", NATO Advanced Study Institute on *Multisensor and Sensor Data Fusion*, Pitlochry, Scotland, June 25 – July 7 2000 (Kluwer Academic Publishers, in press).
2. A. Jouan, P. Valin, and E. Bossé, "Testbed for Fusion of Imaging and Non-Imaging Sensor Attributes in Airborne Surveillance Missions", FUSION 99, 6-8 July 1999, Sunnyvale, CA, Vol. 2, pp. 823-830.
3. P. Valin, A. Jouan, and E. Bossé, "Fusion of imaging and non-imaging sensor information for airborne surveillance", in Proc. on *Sensor Fusion: Architectures, Algorithms, and Application III*, SPIE Aerosense 1999, Orlando, 7-9 April 1999, Vol. 3719, pp. 126-138.
4. P. Valin, H. Aurag, and M. Kjiri, "Initiating 3D air target tracks from 2D naval radar sensor reports", in Proc. on *Signal and Data Processing of Small Targets 2000*, SPIE Aerosense 2000, Orlando, Florida, April 24-28 2000, Vol. 4048, pp. 357-365.
5. M.-A. Simard, P. Valin, and F. Lesage, F. "The ID box: a Multisource Attribute Data Fusion Function for Target Identification", in Proc. on *Sensor Fusion: Architectures, Algorithms and Applications IV*, SPIE Aerosense 2000, Orlando, April 24-28, 2000, Vol. 4051, pp. 180-190.
6. D. Lefebvre, "Classification de cibles navales à l'aide de réseaux de neurones à partir d'images radar à synthèse d'ouverture", M.Sc. thesis, Université de Montréal, Physics Dept. Dec. 1999, 92 pages.
7. P. Valin, Y. Tessier, A. Jouan, "Hierarchical Ship Classifier for Airborne Synthetic Aperture Radar (SAR) Images", in Proc. 33rd Asilomar Conference on *Signal, Systems and Computer*, 24-27 Oct. 1999, Pacific Grove, CA, pp. 1230-1234.
8. P. Valin, and D. Boily, "Truncated Dempster-Shafer Optimization and Benchmarking", in Proc. on *Sensor Fusion: Architectures, Algorithms, and Applications IV*, SPIE Aerosense 2000, Orlando, Florida, April 24-28, 2000, Vol. 4051, pp. 237-246.