Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Information Fusion 4 (2003) 297–308

INFORMATION FUSION

www.elsevier.com/locate/inffus

Cognitive situation and threat assessments of ground battlespaces ☆

Carl G. Looney *, Lily R. Liang

Computer Science Department/171, University of Nevada, Reno, NV 89557, USA

Received 17 July 2002; received in revised form 5 December 2002; accepted 19 February 2003

Abstract

We develop an integrated multi-phase approach to middle and high level data fusion with an application to situation and threat assessments. The method first builds a feature vector for each detected ground target that includes time, position and target class in a particular rectangular geographical area of the battlespace. It then clusters the feature vectors by position using a new robust clustering algorithm and makes an inventory of each cluster as to target classes, counts and posture parameters. Situation assessment is done next via a three-tiered cascaded process of case-based reasoning on cluster attribute records to infer the unit types, sizes, and purposes. These are then fed into our fuzzy belief network that performs inferencing via heuristic belief propagation for threat assessment, that is, it infers the actions and intentions of the enemy. A simple synthetic example demonstrates the process.

Keywords: Information fusion; Clustering; Case-based reasoning; Situation assessment; Fuzzy belief networks

1. Introduction

A profusion of uncertain and often contradictory data from sensors, communications, databases and other sources can be counterproductive [26] to situation awareness by delaying time-critical decision making for operating inside the response cycle of the enemy. *Situation awareness* is a spatial and temporal model of the locations, types, counts, activity, levels of enemy resources [13,18] and echelon organization [12,14] in a battlespace, along with extraneous information on the terrain, weather, routes, population areas, etc. Precise situation awareness enables operational advantage, while an overload of data causes the *fog of war* to become the *glare of war* [26].

Situation assessment (SA) is the ongoing process of inferring relevant information about forces of concern in a military situation [5,6,12,14,15,18,20] to achieve situation awareness. The campaign Commander-in-Chief and staff must use SA to make effective command decisions [17,23]. History shows that it is a key to victory and that the lack of it often leads to surprise and disaster [7,11]. SA includes the analysis of data from sensors and other sources [18] for building situation awareness. SA is a prerequisite for threat assessment (TA) [13], which is the analysis of enemy intentions and capabilities. SA and TA are used for the allocation of resources to targets and other strategy decisions. SA is Level 2 of the DoD Joint Directors of Laboratories Subpanel on Data Fusion [1,14].

Commanders use a visual model of the battlespace for spatial reasoning and decision making [17]. This allows the commander and his forces to operate according to a common perception. Perception is a process [8] of maintaining a model of the state of an environment by combining incoming observations with the previous state and stored data into a coherent description. The ground situation picture is part of the *common operational picture* [23] of the greater battlespace, maintained by the friendly forces using *data fusion* [1].

Section 2 is concerned with the data that commanders need and proposes an integrated multi-step process for assessing situations and threats as part of the overall fusion of data. Section 3 presents a new robust method of clustering the enemy resources by position to obtain cluster attribute records for comparison with cases in a case-base for unit classification. Section 4 describes our use of *cascaded case-based reasoning* to determine type, size and purpose of enemy units from the cluster

☆ Supported by US Army Research Office Grant DAAD19-99-1-0089.
* Corresponding author. Tel.: +1-775-784-4313; fax: +1-775-784-1877.

E-mail addresses: looney@cs.unr.edu (C.G. Looney), liang@cs.unr.edu (L.R. Liang).

1566-2535/$ - see front matter
doi:10.1016/S1566-2535(03)00044-7

attribute data. Section 5 conceptualizes our fuzzy belief networks for threat assessment based on the SA results from the CBR process. Section 6 presents an example for simulation runs on synthetic data, while the analysis and conclusions are presented in Section 7. We use synthetic data because we do not have access to (classified) battlespace data.

2. The problem and cognitive process for assessment needs

The problem in SA and TA is one of how to fuse data from many sources to determine the nature of the enemy, its capabilities and its intended actions in the battlespace region. Most fusion research is on sensor fusion, which is the first level of fusion, because it is better understood [14], while higher levels of fusion are needed to achieve the goal of SA and TA. Our thrust is to use the data fused in Level 1 fusion for Level 2 (SA) and Level 3 (TA) fusion. SA is now being handled in various ways as parts of the overall data fusion [31] to generate a *common operational picture* [23] for visualization. Some of the methodologies being used or investigated include [14] Bayesian probabilities, rule-based systems, neural networks and fuzzy logic, as well as Bayesian belief networks [5], assumption based truth maintenance [6], and statistics and dynamic programming [27].

SA is associated with intelligence preparation for the battlespace (IPB) [34,35], which is the process of providing predictive intelligence to warfighters at the right time for use in planning and executing operations. It provides *information superiority*, or information dominance that allows the collection, control, exploitation and defense of information without effective opposition. The IPB is used to determine [6] the enemy center of gravity (COG) and courses of action (COA), which are parts of SA and TA, respectively. It consists of: (i) defining the battlespace environment; (ii) describing effects of the battlespace; (iii) evaluating the adversary; and (iv) determining the adversary COG. This cycle is repeated throughout a campaign.

The low level data will originate from joint surveillance and target attack radar system (J-STARS), unmanned aerial vehicles (UAVs), piloted aircraft, the new surveillance helicopter (RAH66 Commanche), electronic intelligence and communications intelligence and satellites. These use optics, ladar, infrared (IR) and high frequency radar: the evolutionary trend is to capture images of ground targets for detection, analysis and classification. The data must first be processed for alignment and association [23].

The needs of commanders to make time-critical decisions are: (i) Who is out there? (ii) Where are they? (iii) What is their organizational unit structure and posture? (iv) What are their intentions and goals? (v) How will they try to realize their goals with actions? (vi) What are their resources for sustained effort? and (vii) How do the weather, terrain, position and time-of-day affect the situation?

Our approach to SA and TA is a *cascade* of middle-to-higher level fusion using appropriate methods at different stages. We designate a grid of rectangular geographical areas of the battlespace and for each area we associate a data structure as a file (table) of feature vectors. Each such vector is a record that contains information for a detected and identified (with a certainty value) enemy ground target. The file contains a header of supplemental information from databases (terrain, weather, intelligence reports, and pertinent textual comment). An area may be as small as 10 km by 10 km or much larger. The process is done for each area where nonnegligible enemy resources reside, which may be a single high-threat or high-value target or dozens, hundreds or even thousands of individual targets.

Fig. 1 shows the approach. For SA we first cluster the area target feature vectors by position in the central coordinate system for the battlespace with a new robust clustering algorithm. We then make an inventory of the counts and percentages of the class of targets in each cluster and use these data in three levels of case-based reasoning to determine the unit types, sizes (echelon) and purposes, respectively. For TA we put these results into our new fuzzy belief network, along with weather, terrain and possibly other information to infer the enemy intentions, or courses of action (COA). These processes are described in more detail below.

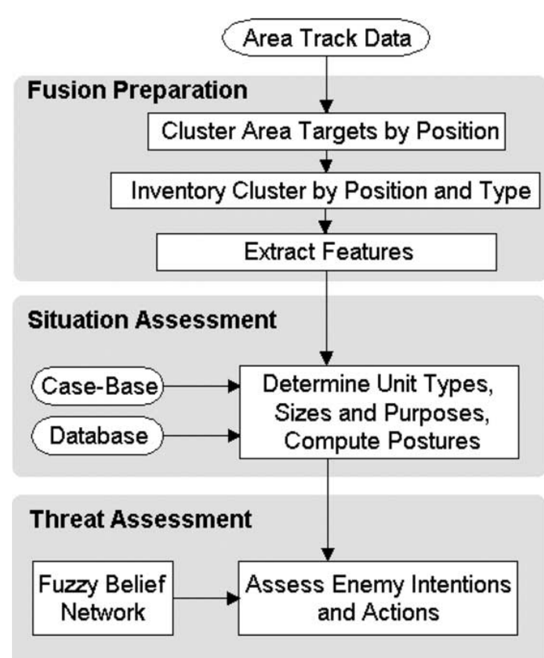

Fig. 1. Cognitive assessment of battlespace.

3. Data clustering

3.1. Target feature vectors

The feature vector for each target contains the time and date of the most recent detection and classification, the position coordinates, the target class (e.g., small, medium or large truck), a belief value for the target class and the status (extent of damage). The belief is a quality, or certainty, value from 0 to 1 where 1 is a full positive identification.

3.2. The centralized k-means clustering algorithm

Clustering yields classification. J.S. Mill wrote in his 1843 work *System of Logic*: "Classification is a contrivance for the best possible ordering of the ideas of objects in our minds... to provide that things shall be thought of in such groups, and those groups in such an order, as will best conduce to the remembrance, and to the ascertainment of their laws".

We cluster the target feature vectors into groups by location for each rectangular area of the battlespace. The information obtained is to be used for SA. Our clustering algorithm is similar in methodology to that of [21], but is simpler and faster in computing the cluster centroids (prototypes) and is also robust. Compared to the k-means algorithm, it provides a much better initial seeding and better clustering with slightly more computation. Bad clustering due to improper seeding is the main fault of the k-means algorithm [21].

The feature vectors for an area of a battlespace are to be updated from the ground target tracks to yield a current set of Q target feature vectors $\{\mathbf{x}^{(q)} : q = 1, \ldots, Q\}$. The clustering of these vectors is done only with respect to position features (x, y) or (x, y, z). The clustering of the Q target feature vectors starts with a relatively large number K of uniformly randomly drawn *seed vectors* $\{\mathbf{z}^{(k)} : k = 1, \ldots, K\}$ as possible centroids of clusters, but thins them to obtain a smaller uniformly distributed set. Starting with the first seed, it thins by eliminating any others closer to it than the threshold T and decrementing K, where T is half the average distance between seeds. Then we check the next available seed in the same way, and so forth. The algorithm now proceeds with the remaining uniformly distributed set of K centroid seeds as given below.

Step 1: For $q = 1$ to Q, assign each feature vector $\mathbf{x}^{(q)}$ to its nearest centroid vector $\mathbf{z}^{(k)}$ by the index $c[q] = k$, increment count of cluster k denoted by *count*[k].
Step 2: Eliminate any centroids not assigned any feature vectors (empty clusters), decrement K accordingly.
Step 3: Compute new centroid for each of the K clusters (average closest 80% to the mean to avoid outliers).
Step 4: Find average of the inter-centroid distances d_{ave} and if any two clusters have centroids closer than $c(d_{\text{ave}})$, $0 < c < 0.5$, then merge them by computing the joint centroid, decrement K, close up indices.
Step 5: Eliminate any cluster k such that $count[k] < p$ (p is user given to eliminate very small clusters).
Step 6: If any change has occurred in any cluster, go to Step 1.

We call the new centroid in Step 2, which ignores outliers, the uniform k-centralized mean (UKCM) of the cluster and give the algorithm the same name. The centroids are more representative than the mean or median, are in effect similar to the *alpha-trimmed mean* [3] and are faster to compute than the *weighted fuzzy expected* values of [21].

3.3. Clustering results: cluster inventories and attributes

When the target clustering of an area is finished, the feature vectors for the ground targets are grouped by location with a centroid for each of its K clusters. At this point we make an inventory of each *Cluster k* with its *count*[k] of target feature vectors shown in Table 1. Each row represents the targets in a target class in a particular cluster. The row belief is the mean of the beliefs over all targets in that row.

From the inventory we make a set of *cluster attribute vectors* that will be used to determine the *unit type*, *size* and *purpose* for each cluster by comparing appropriate features (attributes) with those features in cases to find the best matching ones. The cluster attribute vector for each cluster contains the numbers of targets of each class, the percentages of each class for the cluster and the belief value for each target class.

Table 1
Sample inventory list of an area

Cluster	No. targets	Target class	Time	Location (km)	Mean belief
1	31	4 (Tank)	22:01	(20.92, 34.51)	0.7
1	18	1 (Hvy Truck)	23:52	(20.83, 34.80)	0.8
⋮	⋮	⋮	⋮	⋮	⋮
2	50	8 (Troops)	21:93	(21.71, 32.58)	0.9
⋮	⋮	⋮	⋮	⋮	⋮

4. Situation assessment: inference via case-based reasoning

4.1. The CBR paradigm

Case-based reasoning (CBR) is a paradigm that emulates human reasoning in that it solves new problems by recalling similar problems and their solutions stored in memory [19,25,30]. A *case* is a conceptualization of experience that represents knowledge and contains a past lesson that includes the context for the case in which the lesson is used [2,9,19]. A case typically is composed of the parts: (i) the *problem* that describes the state of the environment when the case occurred; (ii) the *solution* to the problem part; and (iii) the *outcome* (or *success metric*) of the state of the environment [2] after the solution was applied.

A given input problem is compared with the case problem parts in the case-base to retrieve cases with the best match so their solution parts can be used. Fig. 2 shows the CBR paradigm and Fig. 3 displays the CBR cycle that learns by accumulating experience. The cycle of CBR for a new input problem is: (i) *retrieve* the most similar cases from the case-base; (ii) *re-use* them to attempt to solve the problem; (iii) *revise* them if necessary to better fit the new problem; and (iv) *retain* the new solution with the new problem as a new case if it is shown to be a good solution.

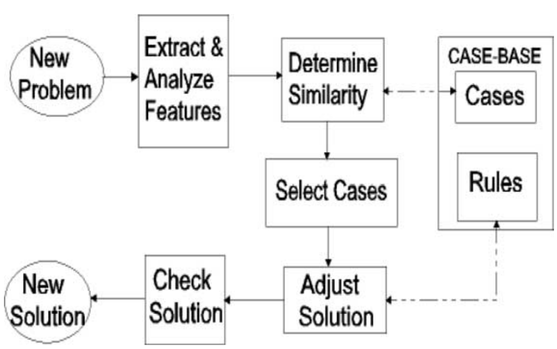

Fig. 2. The problem solving paradigm.

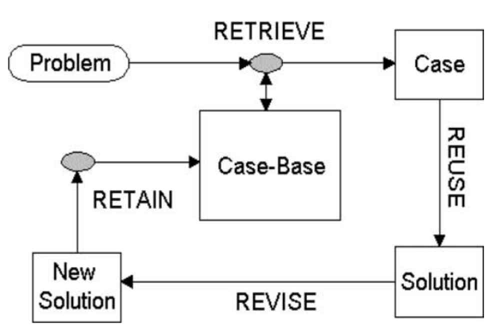

Fig. 3. The CBR cycle.

CBR has been used successfully in planning, design, diagnosis, assembly and scheduling [19]. For the purpose of SA, we design a series of three levels of case based reasoning to get the unit type of the cluster (infantry, artillery, etc.), size (platoon, company, battalion, etc.) and purpose (attack, defend, feint, raid, retreat, reserve, etc.), respectively, for later use in threat assessment.

4.2. High-level design

One simple case-base is implemented for each of the three levels of reasoning. The cluster attribute vector of each cluster is processed via all three levels, one at a time, until an entire area of clusters is done. Fig. 4 presents the process flow. The process is repeated for each area where targets have been detected, classified and clustered.

The *first level* of CBR compares the target class percentages of the cluster with those of cases in the first-level case-base, and gets the unit type (i.e., armor, infantry, artillery, SAM site, etc.) of the cluster as the solution. This result is taken as input for the second-level CBR, which additionally requires the total number of targets in the cluster. This process compares that data with cases in the *second-level* case-base to get the unit size (i.e., platoon, company, battalion, etc.) of the cluster as the solution. Both unit type and size are then fed into the *third-level* CBR with posture parameters of the cluster targets, so the purpose of that cluster can be deduced.

4.3. Case design and storage

Case storage is important for CBR efficiency, so its design should support and simplify the retrieval process. For the case-base of each level, we use a dynamic memory model [19,30], where each instance of the basic unit is a case. The idea is to organize specific cases that

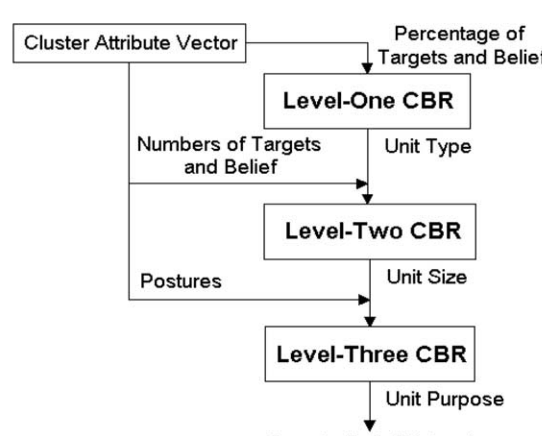

Fig. 4. High-level CBR procedure.

share similar properties under a more general structure, i.e., a *generalized episode* (GE).

The memory is *dynamic* in that similar parts of two or more cases are dynamically generalized into a new GE, the cases being indexed under the GE by their differences [33]. We implement GEs via all three levels of case-bases.

For the *first-level case-base*, which is used to determine the unit type, the problem part is a record that contains the percentage of each target class in a particular cluster. The solution part of it is the type of enemy unit that the cluster represents (infantry, mechanized infantry, armor, artillery, etc.). A new cluster is assigned a unit type by comparison with the case-base records to find the most similar case. The similarity depends not only on the composition of the incoming cluster and the case being compared, but also on the importance of the target classes to the unit type. An *importance weight* is applied to the percentage of each target class in a cluster. The weights are standardized by dividing by the sum of weights.

The *second-level case-base* is used to determine the unit echelon, or size (platoon, company, battalion, brigade, regiment, division). The problem part of the cases includes the unit type, the raw numbers of targets of different target classes and the total numbers of vehicles and weapons (other than small arms), while its solution part is the *unit size*.

Similarly, the *third-level case-base* uses a problem part that has the unit type, unit size and the cluster posture parameters, while the solution part is the unit purpose that may be attack, defend, feint, raid, reserve, wait, or unknown. We can not determine the purpose of a unit (cluster) without information on the other clusters in the same area. Also, posture of a unit depends on the types and sizes of the other units, as well as their relative locations. Thus the first and second level CBR processing of the clusters in the same area is done before the third level. Importance weights are also applied when comparing the cluster data with the cases because the features are not weighted equally. There are three posture parameters: (i) distance from the enemy cluster COG to the frontier of contention; (ii) the mass parameter (total units per square kilometer); and (iii) the mixture of armor, artillery, SAMs, mechanized infantry, regular infantry, etc.

4.4. Case retrieval

There are several well-known case retrieval methods, of which the most useful ones are nearest neighbor, induction, knowledge guided induction and template retrieval [33]. Here we use a hybrid retrieval strategy. It combines the nearest neighbor method with the template retrieval method, and thus solves the retrieval time problem of the former while achieving flexibility in matching that the latter method does not have. Three smaller case-bases also yield higher efficiency than one large one.

Template retrieval is similar to common database queries in that it returns cases that satisfy certain parameter constraints. In this method, the new problem is parsed to extract the more important features to be used as a template. Any case in the case-base that matches the template is selected for further checking and the others are skipped. We use it at the beginning of the retrieval process to narrow down the space for further search.

After the selected cases have been fetched, the nearest *neighbor method* is applied by computing the *similarity* between the input problem and the stored cases. The similarity uses the proximity, the degree of belief and the importance weight over all problem attributes. The more important an attribute is, the more it influences the case similarity. At each level we calculate the respective similarity for unit type, size and purpose that are respectively designated by S_t, S_s and S_p.

Let there be K clusters for an area. We start with the attribute vector for the kth cluster, select the next jth record from the *unit-type* case-base of J case records and then sum the relative weighted difference magnitudes over all N target class attributes in the cluster. The *type* similarity measure is

$$S_t(k,j) = \frac{\sum_{n=1,N}(w_{nj})\{\{|P_{nk}-P_{nj}|/(P_{nj}+1)\}+1\}^{-1}(B_{nk})}{\sum_{n=1,N}(w_{nj})(B_{nk})} \tag{1}$$

where w_{nj} is the importance weight of nth target class for the type stored in the jth case record, P_{nk} the proportion of the nth target class in the kth cluster, P_{nj} the proportion of the nth target class in the jth case record and B_{nk} the mean belief of the nth target class in the kth cluster. Similar notations hold for the *size* and *purpose* similarities given below.

We use Q_{nj} and Q_{nk} as the quantities for size similarity in place of proportions. T_k and T_j are the total numbers of targets, respectively, for cluster k and case record j.

$$S_s(k,j) = \frac{(1/2)\sum_{n=1,N}\{\{|Q_{nk}-Q_{nj}|/(Q_{nj}+1)\}+1\}^{-1}(B_{nk})}{\sum_{n=1,N}(B_{nk})} + (1/2)\{[|T_k-T_j|/(T_j+1)+1]^{-1} \tag{2}$$

We use p_{rk} and p_{rj} as the posture parameters for the rth posture of the kth cluster and jth case record, respectively. The importance weights $\{u_{rj}\}$ in Eq. (3a) are constants input by the user.

$$S_p(k,j) = (1/R)\sum_{r=1,R}(u_{rj})(V_r) \tag{3a}$$

$$V_r = 1, \quad \text{if } p_{rk} = p_{rj}, \quad \text{else } V_r = 0 \tag{3b}$$

Fig. 5 shows a generalized CBR flow diagram for any of the three levels, where a particular cluster is used on

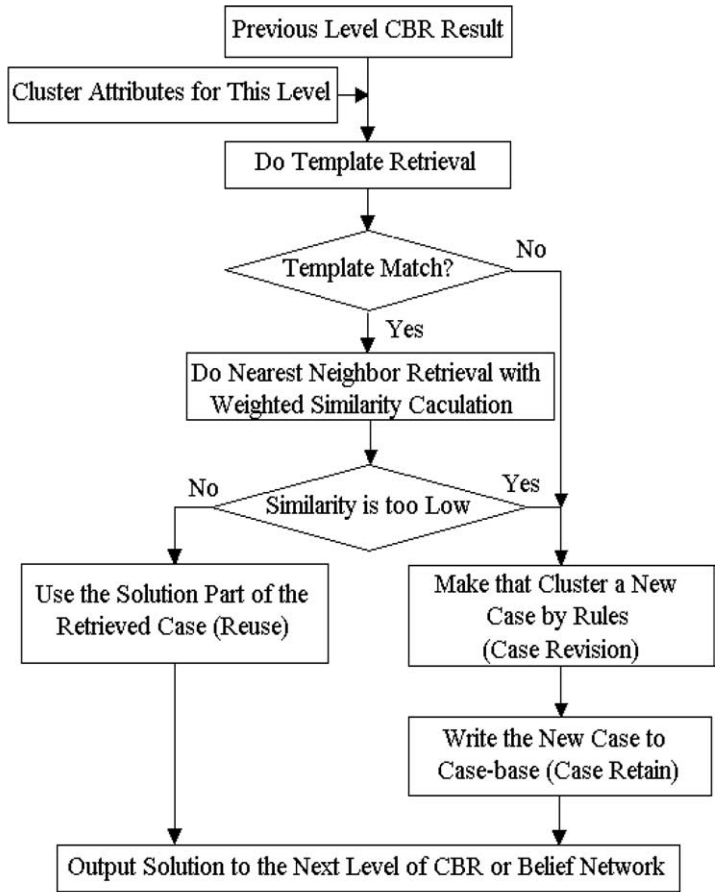

Fig. 5. The CBR flow diagram.

each run with the cluster features being used as the problem to match against the problem parts of the cases. At any level, the case with the greatest weighted similarity is considered the most similar case, and its solution is retrieved and used as the solution of the input problem. That is, the unit type, size and purpose of the retrieved cases at different levels will be used for the new cluster and fed accordingly to the next level of CBR or to the threat assessment procedure.

The calculated similarity is considered to be the *case belief*. Both the solutions and their case beliefs will be output as the result of the case-base reasoning for each level. When there is no matching result in the template matching step (the similarity is too low), the problem is considered to be a *new* case for which a solution must either be entered by a human or be deduced by rules and checked for success in either case.

5. Threat assessment: inference via fuzzy belief networks

5.1. Belief networks

A *Bayesian belief network* (BBN) is an acyclic directed graph [4,10,16], where the nodes represent (discrete) random variables and the arrows represent causal influences. The prior probability tables (PPTs) at the root nodes and conditional probability tables (CPTs) elsewhere are required to compute the joint probability distribution (jpdf) of all variables. From a given set of *observation* variables, the likelihoods of the outcomes of a set of *query* variables are to be determined. The jpdf is needed for such inferencing.

Fig. 6 shows a simple BBN of nodes A, B, C, D and E that has nodal outcomes that are true (T) or false (F). We let Ta and Fa be the respective *true* and *false* outcomes for A, Tb and Fb be the outcomes for B, etc. For examples, we see from Fig. 6 that $\mathrm{P[C = Fc \mid E = Te] = 0.3}$ and $\mathrm{P[E = Te \mid A = Ta,\ B = Fb] = 0.4}$.

Let $p(X_k)$ denote the set of parent nodes of the random variable X_k, so the jpdf of the random variables is given by Eq. (4). Assuming that each variable X_k is conditionally independent of its non-descendants when given the outcomes of its parents [29], the BBN of Fig. 6 can be simplified per Eq. (5).

$$\mathrm{P}[X_1, X_2, \ldots, X_K] = \mathrm{P}[X_1|p(X_1)]\mathrm{P}[X_2|p(X_2)]\ldots\mathrm{P}[X_K|p(X_K)] \tag{4}$$

$$\mathrm{P}[A,B,C,D,E] = \mathrm{P}[A]\mathrm{P}[B]\mathrm{P}[E|A,B]\mathrm{P}[C|E]\mathrm{P}[D|E] \tag{5}$$

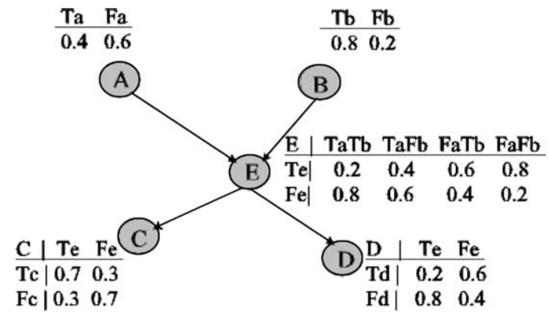

Fig. 6. A simple BBN.

With this simplification we can compute the jpdf for all possible outcomes (A,B,C,D,E) of the BBN in Fig. 6, which are the $2^5 = 32$ events. Such exponential combinations make BBNs NP-hard. An event is any outcome in {TaTbTcTdTe, TaTbTcTdFe,...,FaFbFcFdFe}. For example, from the CPTs of Fig. 6 and Eq. (5) we obtain

$$\begin{aligned} &\mathrm{P[Ta, Fb, Fc, Td, Te]} \\ &\quad = \mathrm{P[Ta]P[Fb]P[Te|Ta, Fb]P[Fc|Te]P[Td|Te]} \\ &\quad = (0.40)(0.20)(0.40)(0.30)(0.20) = 0.000192 \end{aligned} \tag{6}$$

An example of inferencing is where we let E be the decision variable that we wish to query ($E = \mathrm{Te}$?) based on the observed outcome $C = \mathrm{Tc}$. The Bayesian rule provides that

$$\begin{aligned} &\mathrm{P}[E = \mathrm{Te}|C = \mathrm{Tc}] \\ &\qquad = \mathrm{P}[C = \mathrm{Tc}|E = \mathrm{Te}]\mathrm{P}[E = \mathrm{Te}]/\mathrm{P}[C = \mathrm{Tc}] \end{aligned} \tag{7}$$

$\mathrm{P}[C = \mathrm{Tc}|E = \mathrm{Te}]$ is provided by the CPT at C, but the two marginal probabilities $\mathrm{P}[C = \mathrm{Tc}]$ and $\mathrm{P}[E = \mathrm{Te}]$ must be computed. To compute $\mathrm{P}[E = \mathrm{Te}]$, we sum the probabilities of all 16 combinations of outcomes ABCD, where $E = \mathrm{Te}$ is fixed. We similarly compute $\mathrm{P}[C = \mathrm{Tc}]$.

5.2. Why heuristic fuzzy belief networks?

While Bayesian probabilities are suitable for updating belief networks and allow for forward and backward influences to be computed, the (usually subjective) values for the PPTs and CPTs are difficult to estimate and the jpdf is NP-hard to compute. Two other models that could be used for updating are: (i) fuzzy logic; and (ii) theory of evidence. We choose fuzzy logic because it is heuristic, direct, intuitive, easy to tune and has low order polynomial linear time complexity. Other approaches that do not use fuzzy logic are in [6,32]. But fuzzy logic is more heuristic, faster, easier to understand, more common-sensical and the forward–backward directions are trivially handled (see below).

For this application of our [22] fuzzy belief network (FBN) to threat assessment, we use the enemy intentions as the query variables at the root nodes. Each intention requires certain actions at the children of the root nodes, and each action requires a particular mixture of force units in certain areas at the leaf nodes. The SA data are the observations that provide the unit locations, types, sizes, postures and purposes, which match up with certain of the leaf node units in the FBN. The extent to which the outcome of a variable influences the beliefs of outcomes of its parent or child variables is determined by its fuzzy set membership function (FSMF) for the propagation along an arrow. There is a FSMF for each direction along an arrow, which establishes two-way conditional influences.

Fig. 7 shows the FSMF for propagating the belief influence that the variable X is present. We model these monotonically increasing (*Tsukamoto*) fuzzy set membership functions by *sigmoid functions* (e.g., see [24]) to propagate fuzzy beliefs that particular units are present in the battlespace, where the observations are beliefs of the presence of units given by the SA.

The propagated fuzzy belief y is given by the sigmoid function with *center c*, *rate a* and fuzzy belief x as provided by the sigmoid of Eq. (8) that attains values between 0 and 1. It is also continuously differentiable.

$$y = g(x) = 1/\{1 + \exp[-a(x - c)]\} \tag{8}$$

Fuzzy logic is extremely flexible (see [28]). A rule has the form (*A is present*) $[f_A]$ *AND* (*B is present*) $[f_B]$ => (*C is present*) $[f_C]$, where the antecedents A and B have respective fuzzy beliefs f_A and f_B that influence the fuzzy belief f_C that C is present.

A variety of fuzzy logic methods are available to determine f_C. A well known method is $f_C = \min\{f_A, f_B\}$, but others use specific ones where $\min\{f_A, f_B\} < f_C < \max\{f_A, f_B\}$ [28]. The product $f_A * f_B$ does not satisfy this criterion. A tradeoff is to employ weights via

$$f_C = (w_A)(f_A) + (w_B)(f_B) \tag{9}$$

where the weights are standardized to sum to unity. The influence f_C must then be logically combined with the current fuzzy belief $f_{C^\wedge}$ of the presence of C to obtain a

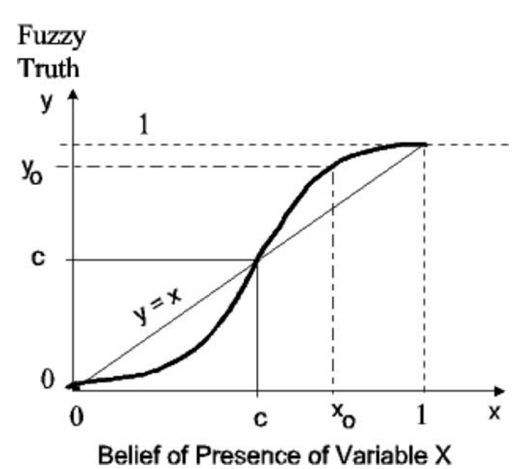

Fig. 7. A Tsukamoto FSMF for X.

new one $f_{Cnew} = F[f_{C^\wedge}, f_C]$ for some logic function $F[\quad]$. There are various ways to do this and here we used a proportional increase or decrease from $f_{C^\wedge}$.

6. Simulation experiments: design and results

6.1. The simulation of SA via CBR

For the purpose of showing how the integrated system works it suffices to use an example of four types of units in a mix of: (i) one heavy infantry battalion as Unit *A*; (ii) an infantry brigade *B*; (iii) an armored battalion *C*; (iv) an artillery company *D*; (v) an infantry battalion *E*; and (vi) an armored company *F*. Units *A*, *B*, *C* and *D* have a posture of being close to the frontier, close to each other (massed), in a mixture that supports an attack across the frontier. The postures and sizes of Units E and F support a reserve status for offense or defense.

For simplicity, we do not distinguish between types of tanks or trucks, such as heavy, medium or light tanks. Fig. 8 shows an area ground situation (of clusters) for a test of the SA and TA processes. The area where the targets are detected is a 15 km by 15 km area (9.3 miles on a side) with the frontier running along the Eastern edge. The postures (relative positions) are shown in Fig. 8. The test situation shown in the figure is described in Table 2.

There are 6 clusters in Table 2 that represent the units shown. Associated with each set of targets in a certain class is the average belief in the classification of the targets, which is the mean of the beliefs of the individual targets. For each cluster there are N different target classes that are important for defining the type of unit that cluster represents.

Table 3 displays part of a case-base of percentages of the various weapons classes for the particular unit type, where unit type is the solution. This is the *first level* case-base and all of the data used here are synthetic. The importance weights of the target classes are not shown but were real values between 0 and 1.

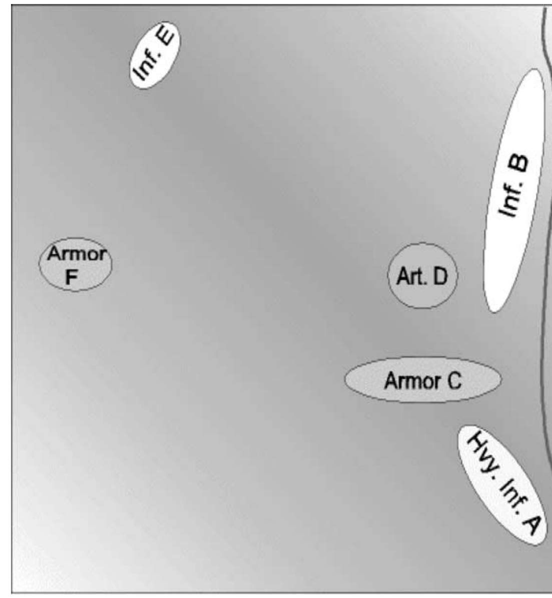

Fig. 8. Area with infantry, armor & artillery.

The different attributes are weighted differently in the three different case bases. In determining the size of the unit, the total number of target vehicles is as important as all of the other attributes combined. The weights for the third level of case-based reasoning are: (i) Posture 1 weight is 0.7; (ii) Posture 2 weight is 0.8; and (iii) Posture 3 weight is 0.9. The results are described below, where the SA results of the CBR are fed into the FBN. The weather and terrain were not included in these runs.

6.2. Simulation of threat assessment via an FBN

We use a simple synthetic example of a network here (not designed by military experts) that suffices to show

Table 2
Test case: cluster attribute vectors/beliefs

Area no. 1							
Cluster no.	1 (A)	2 (B)	3 (C)	4 (D)	5 (E)	6 (F)	Totals
No. trucks/belief	55/0.8	50/0.7	15/0.8	12/0.8	30/0.7	8/0.8	170
% Trucks	32	32	16	26	37	35	
No. SP guns/belief	12/0.7	10/0.7	0	22/0.6	6/0.7	0	50
% SP guns	07	06	00	48	07	00	
No. APCs/belief	60/0.7	50/0.6	12/0.7	0	20/0.7	1/0.7	143
% APCs	34	32	13	00	25	04	
No. tanks/belief	20/0.9	25/0.8	55/0.8	0	10/0.9	14/0.8	124
% Tanks	12	16	60	00	12	61	
No. ATGMs/belief	21/0.6	20/0.5	10/0.6	12/0.7	12/0.6	0	75
% ATGMs	12	13	11	26	15	00	
No. AA SAMs/belief	6/0.6	2/0.7	0	0	3/0.6	0	11
% AA SAMs	03	01	00	00	04	00	
Totals	174	157	92	46	81	23	573

SP: self-propelled, APC: armored personnel carrier. ATGM: anti-tank guided missile, SAM: surface-to-air missile.

Table 3
First-level case-base sample record

% of Trucks	% of SP guns	% of APCs	% of Tanks	% of ATGMs	% of AA SAMs	Unit type
29	7	36	13	13	1	2 (hvy. inf.)
31	7	31	15	13	4	3 (infantry)
22	0	8	57	14	0	1 (armored)
29	41	0	0	29	0	4 (artillery)
52	6	24	13	0	5	3 (infantry)
37	0	0	63	0	0	1 (armored)

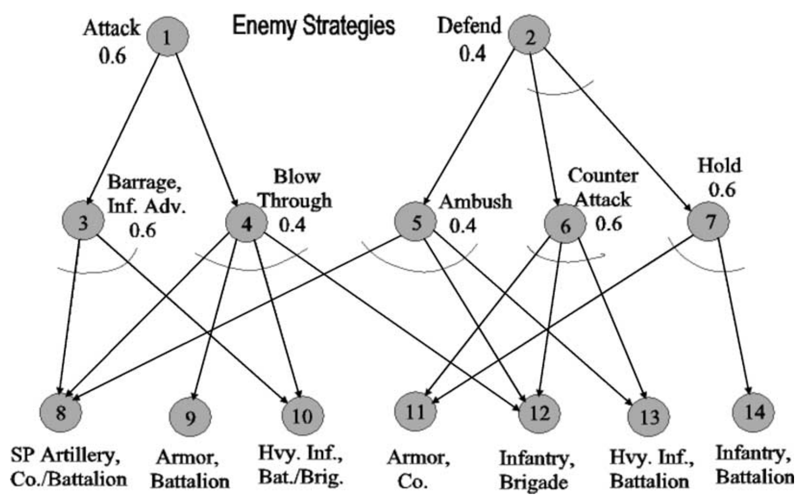

Fig. 9. The FBN for the example ground situation.

the working of the methodology. Fig. 9 shows that the enemy is expected to be planning either to *attack* with prior belief 0.6 or to *defend* with belief 0.4 along the frontier of Fig. 8. The attacks can come either as a *blow-through* or an *artillery barrage* and *infantry advance*. The blow-through requires sufficient armor and heavy motorized infantry that are supported by self-propelled artillery and regular infantry as shown in Fig. 9. On the other hand, a more cautious attack would require artillery and heavy (motorized) infantry. The a priori beliefs, to be given from the expertise of the command staff, are shown.

Because FBNs can be complex, we devise a simple and efficient data structure for the influence graph with its deductive (forward) influences and its abductive (backward) influences. This example is synthetic and enemy strategies and actions depend on the actual enemy battle doctrine.

For a real world situation, the command staff would set up the belief network. The following influences are from Fig. 9, where the integers are the nodal numbers.

$$\begin{aligned}\text{Deductive Influences}: &\ 1 => 3; \quad 1 => 4; \quad 2 => 5; \\ &\ 2 => 6,7; \quad 3 => 8,10; \\ &\ 4 => 8,9,10,12; \\ &\ 5 => 8,12,13; \\ &\ 6 => 11,12,13; \quad 7 => 11;14\end{aligned}$$

$$\begin{aligned}\text{Abductive Influences}: &\ 3 => 1; \quad 4 => 1; \quad 5 => 2; \\ &\ 6,7 => 2; \quad 8,10 => 3; \\ &\ 8,9,10,12 => 4; \quad 8,12,13 => 5; \\ &\ 11,12,13 => 6; \quad 11,14 => 7\end{aligned}$$

The original set of influence rules is stored in the file *FBNfile.dat* as rows, that is, strings of digits and spaces ended by a carriage return (CR). We write at the beginning of each row the number of integers to follow in that row, so we have a file that is short-hand for linked lists and is more efficient. For example, the influence rule $1 = >3$ is represented by the line 2, 1, 3 (CR), while $4 = >8$, 9, 10, 12 is represented by 5, 4, 8, 9, 10, 12 (CR).

Our program *setNodeLinks.php* is interpreted by the *PHP interpreter* under the *Apache* Web (http) server and the results are sent to a browser such as *Netscape Navigator* as *HTML* for display. Our program parses these rules and creates the abductive influence rules for the backward influencing process. For example, the abductive influence rule $3 = >1$ is written as the file line 2, 3, 1 (CR) and the more complex 8, 9, 10, $12 = >4$ is stored as the line 5, 8, 9, 10, 12, 4 (CR) (the first stored integer gives the number of node numbers to follow). Only the last number in the row is a rule consequent node (the others are antecedents).

The SA output data from the CBR process are written to the file *SAfile.dat* as rows

$k, T(k), B_T(k), S(k), B_S(k), P(k), B_P(k)$

where k is the cluster number, $T(k)$ the type for Unit k, $B_T(k)$ the belief of $T(k)$, $S(k)$ the size of Unit k, $B_S(k)$ the belief of $S(k)$, $P(k)$ the purpose of Unit k and $B_P(k)$ the belief of $P(k)$.

To activate the FBN antecedent units (leaf nodes) for the abductive influence rules, we must match them with the SA output (cluster) units. These FBN antecedent units are stored in the file *unitsFBNfile.dat* that is to be written whenever a new FBN structure file is written. The rows for *unitsFBNfile.dat* contain

$n(m), T(m), S(m), P(m)$

Here m is the unit number, $n(m)$ is the nodal number of the mth unit in the FBN graph, $T(m)$ is the type, $S(m)$ is the size and $P(m)$ is the purpose of Unit m.

The high level FBN algorithm for TA is shown below and is run from a Web browser by typing in the URL http://ultima.cs.unr.edu/fzBN/fbn.htm to bring up this FBN (http://ultima.cs.unr.edu/fzBN2/fbn.htm brings up a different FBN for an experiment).

Step 1: On click of the *Run* button, read the file *FBNfile.dat* to get the network structure and then run *setNodeLinks.php* to create the abductive influence rules, read *unitsFBNfile.dat* to get the units (leaf node) data for the fuzzy belief network and read *beliefFile.dat* to get the initial beliefs of the non-leaf nodes.

Step 2: Read the files *SAfile.dat* to get the current output data of the CBR situation assessment process and look for the best matches with the units in *unitsFBNfile.dat*. Set the belief accordingly for each FBN unit (leaf node) that matches an SA unit to initialize the FBN with initial observations and beliefs.

Step 3: Process the abductive influence rules by use of FSMFs to adapt the beliefs of the *action* nodes, then adapt the beliefs of the *intention* (root) nodes and output the beliefs of these query nodes.

Step 4: Repeat forever.
if (file *SAfile.dat* is updated) then go to Step 2 //new situation
if (*FBNfile.dat* is updated) then go to Step 1 //new network.

6.3. The simulation results

The situation assessment (unit) data are shown in Table 4 and the leaf node unit data are given in Table 5 for the determination of the beliefs of the matched leaf node units. The rules for abductive belief influence propagation and original influences from Fig. 9 are given in Table 6.

Table 4
The SA unit data

Clstr no.	Type	Belief	Size	Belief	Purpose	Belief
1	3	92	4	81	1	80
2	3	92	4	75	1	80
3	1	84	3	84	1	80
4	4	89	2	89	1	80
5	3	88	3	75	4	80
6	1	83	2	81	4	80

Table 5
The FBN leaf node (unit) data

Unit	Node	Type	Size	Purpose
1	8	4	3	1
2	9	1	3	1
3	10	3	4	1
4	11	1	2	4
5	12	3	4	2
6	13	2	3	4
7	14	3	3	1

Table 6
The deductive and abductive influence rules for fuzzy belief propagation

Rule no.	Deductive rule	Abductive rule
1	1 => 3	3 => 1
2	1 => 4	4 => 1
3	2 => 5	5 => 2
4	2 => 6, 7	6, 7 => 2
5	3 => 8, 10	8, 10 => 3
6	4 => 8, 9, 10, 12	8, 9, 10, 12 => 4
7	5 => 8, 12, 13	8, 12, 13 => 5
8	6 => 11, 12, 13	11, 12, 13 => 6
9	7 => 11, 14	11, 14 => 7

Table 7 gives the results of firing the fuzzy belief influence rules through two cycles. L levels of nodes require $L-1$ cycles of rule processing, and here $L=3$, with levels 0 (roots), 1 (root children) and 2 (leafs).

Table 8 shows the final beliefs of all nodes after the rule processing has been completed. The beliefs of the root nodes are standardized to sum to unity, which allows them to be considered as likelihoods. For the other nodes the raw beliefs as observed and adjusted are given. If a new set of observations were to be entered then non-leaf nodes would need to be reset to the initial values, or new initial values could be given.

7. Analysis and conclusions

To solve the problem of fusion for SA and TA, we selected a sequence of methodologies. We developed the UKCM clustering algorithm that uses a profuse uniformly random seeding and thinning to prevent bad

Table 7
The results of firing the abductive influence rules

Rule update loop: 1
Node 1, belief is 0.6, Rule 1 was NOT fired!
Node 1, belief is 0.6, Rule 2 was NOT fired!
Node 2, belief is 0.4, Rule 3 was NOT fired!
Node 2, belief is 0.4, Rule 4 was NOT fired!
= > Node 3, weighted average of antecedent beliefs of Rule 5: 0.8656
Fuzzy value is: 0.961421824
Node 3 has old belief: 0.6
Node 3 has new belief: 0.871066368
Node 4, belief is 0.4, Rule 6 was NOT fired!
Node 5, belief is 0.4, Rule 7 was NOT fired!
Node 6, belief is 0.6, Rule 8 was NOT fired!
Node 7, belief is 0.6, Rule 9 was NOT fired!
Rule update loop: 2
= >Node 1, weighted average of antecedent beliefs of Rule 1: 0.871066368
Fuzzy value is: 0.96411856815886
Node 1 has old belief: 0.6
Node 1 has new belief: 0.87308892611915
Node 1, belief is 0.87308892611915, Rule 2 was NOT fired!
Node 2, belief is 0.4, Rule 3 was NOT fired!
Node 2, belief is 0.4, Rule 4 was NOT fired!
Node 3, belief is 0.871066368, Rule 5 was NOT fired!
Node 4, belief is 0.4, Rule 6 was NOT fired!
Node 5, belief is 0.4, Rule 7 was NOT fired!
Node 6, belief is 0.6, Rule 8 was NOT fired!
Node 7, belief is 0.6, Rule 9 was NOT fired!

Table 8
The final fuzzy beliefs of all nodes

Node 1: Final belief = 0.68580356659032 (standardized from 0.83731)
Node 2: Final belief = 0.31419643340968 (standardized from 0.40000)
Node 3: Final belief = 0.871066368 (not standardized)
Node 4: Final belief = 0.4 (not standardized)
Node 5: Final belief = 0.4 (not standardized)
Node 6: Final belief = 0.6 (not standardized)
Node 7: Final belief = 0.6 (not standardized)
Node 8: Final belief = 0.8666 (not standardized)
Node 9: Final belief = 0.8296 (not standardized)
Node 10: Final belief = 0.8646 (not standardized)
Node 11: Final belief = 0.8178 (not standardized)
Node 12: No change from 0.1 (not standardized)
Node 13: No change from 0.1 (not standardized)
Node 14: No change from 0.1 (not standardized)

clusters. This clustering of the targets by position allowed us to perform SA by making an inventory of the clusters and then using our technique of cascaded case-based reasoning for the recognition of the type, size and purpose. Our fuzzy belief network then processed the SA data to modify the beliefs of the query nodes according to the observations and their influence links that propagate the belief influences between nodes very quickly by sigmoid fuzzy set membership functions. A Bayesian belief network would require data mining and estimators to obtain the conditional probabilities and then would require exponential computing time.

The example on which we ran the simulation was synthetic and fairly simple, whereas real world applications can be complex. However, it demonstrated the possibilities for larger structures and with real world data known to military experts. In this simulation the example required only propagating the beliefs abductively (backwards), but we have other examples where it works both ways. Future work will involve more complex structures and inferencing for detection, classification and decision making.

The results here showed that a multistage process of data fusion can be used to infer information for decision making. It is clear that the fuzzy belief network can also be used for classification based on fuzzy beliefs of the degree of certain features (variables) represented by nodes. Our fuzzy belief network could have been used with a more complex network structure before the Iraqi invasion of Kuwait to ascertain the intentions of the Iraqi government. Such intentions were incorrectly surmised by the US State Department in 1990 based on their own cultural values, but were accurately determined by the US Defense Intelligence experts based on the demonstrated values of the ruling party in Iraq.

References

[1] Data Fusion Development Strategy Panel, Functional Description of the Data Fusion Process, Office of Naval Technology, Washington, DC, November 1991.

[2] A. Aamodt, E. Platz, Case-based reasoning: foundational issues, methodological variations and system approaches, IEEE AI Commun. 7 (1) (1994) 39–59.

[3] J.B. Bednar, T.L. Watt, Alpha-trimmed means and their relationship to median filters, IEEE Trans. Acoust., Speech Signal Process. 32 (1) (1984) 145–153.

[4] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: E. Horvitz, F. Jensen (Eds.), Proc. Twelfth Annual Conf. on Uncertainty in Artificial Intelligence, Portland, OR, Morgan Kaufmann, 1996, 115–123.

[5] K.C. Chang, Z. Tian, Efficient inference for mixed Bayesian networks, Proc. Fus. 1 (2002) 527–534.

[6] J.W. Choi, J.W. Joo, D.L. Cho, Situation/threat assessment fusion systems, Proc. Fus. 2 (2002) 1374–1380.

[7] E.A. Cohen, J. Gooch, Military Misfortunes, the Anatomy of Failure in War, Vintage Books, Random House, Inc, New York, 1991.

[8] J.L. Crowley, Principles and techniques for sensor data fusion, in: J.K. Aggarwal (Ed.), Multisensor Fusion for Computer Vision, Springer-Verlag, 1993, pp. 15–36.

[9] B.S. David, Principles for case representation in a case-based aiding system for lesson planning, in: Proc. Workshop on Case-Based Reasoning, Washington, DC, May 1991.

[10] M.J. Druzdzel, Five useful properties of probabilistic knowledge representations from the point of view of intelligent systems, Fundamenta Informaticae 30 (3-4) (1997) 241–254.

[11] J.F. Dunnigan, How to Make War: a Comprehensive Guide to Modern Warfare for the Post Cold-War Era, 3rd ed., Morrow and Co., Inc., New York, 1993.
[12] P.G. Gonzalves, G.J. Rinkus, Intelligent fusion and asset manager processor, in: Proc. IEEE Info. Technology Conf., 15–18, 1998.
[13] P. Gonsalves, R. Cunningham, N. Ton, D. Okon, Intelligent threat assessment processor (ITAP) using genetic algorithms and fuzzy logic, in: Proc. Fusion 2000, Paris, July 2000, ThB1-18–ThB1-24, 2000.
[14] M.L. Hinman, Some computational approaches for situation assessment and impact assessment, Proc. Fusion 2002, 1 (2002) 687–693.
[15] W. Horrey, C. Wickens, Supporting situation assessment through attention guidance: a cost-benefit and depth-of-processing analysis, in: Proc. 45th Annual Meeting of the Human Factors and Ergonomics Society, Santa Monica, CA, 2001.
[16] F.V. Jensen, Bayesian Networks and Decision Graphs, Springer-Verlag, Inc., New York, 2001.
[17] J.P. Kahan, D. Robert Worley, C. Stasz, Understanding Commanders Information Needs, Rand/Arroyo Center, Santa Monica, 2000.
[18] D. Kettani, J. Roy, A qualitative spatial model for information fusion and situation analysis, in: Proc. Fusion 2000, Paris, France, vol. I, July 2000, TuD1-16–TuD1-23, 2000.
[19] J.L. Kolodner, Maintaining organization in a dynamic long-term memory, Cognit. Sci. 7 (4) (1983) 243–280.
[20] K.B. Laskey, B. DAmbrosio, T.S. Levitt, S. Mahoney, Limited rationality in action: decision support for military situation assessment, Mind Mach. 10 (2000) 53–77.
[21] C.G. Looney, Interactive clustering and merging with a new fuzzy expected value, Pattern Recognit. 35 (11) (2002) 2413–2423.
[22] C.G. Looney, L.R. Liang, Inference via fuzzy belief networks, in: Proc. ISCA CAINE Int. Conf., San Diego, 25–28, 2002.
[23] C.G. Looney, Exploring fusion architecture for a common operational picture, J. Information Fus. 2 (2001) 251–260.
[24] C.G. Looney, Pattern Recognition using Neural Networks, Oxford University Press, NY, 1997.
[25] Ramon Lopez de Mantaras, E. Plaza, Case-based reasoning: an overview, AI Commun. 10 (1997) 21–29.
[26] H.S. Marsh, Beyond Situation Awareness: The Battlespace of The Future, ONR White Paper, March 2000.
[27] D. McMichael, A statistical approach to situation assessment, in: Proc. Fusion 99, Sunnyvale, CA, July 1999.
[28] J.M. Mendel, Uncertain Rule-based Fuzzy Logic Systems, Prentice-Hall, Upper Saddle River, NJ, 2001.
[29] J. Pearl, Fusion, propagation and structuring in belief networks, Artif. Intell. 29 (3) (1986) 241–288.
[30] R. Schank, Dynamic Memory: A Theory of Reminding and Learning in Computers and People, Cambridge University Press, Cambridge, UK, 1982.
[31] P. Stiles, M. Hoffmann, Demonstrated value of data fusion and situation assessment, presented paper at American Helicopter Society Avionics and Crew Systems Technical Specialists Meeting, Philadelphia, September, 1999.
[32] S. Nadkarni, P.P. Shenoy, A Bayesian network approach to making inferences in causal maps, Eur. J. Operat. Res. 128 (2001) 479–498.
[33] I. Watson, F. Marir, Case-based reasoning: a review, at http://www.ai-cbr.org/classroom/cbr-review.html.
[34] Information Operations, Air Force Doctrine Document (AFDD) 2–5, 41–42, August 1998.
[35] Intelligence Preparation of the Battlefield, Army Field Manual 34-130, July 1994.