Available online at www.sciencedirect.com
ScienceDirect
Information Fusion 8 (2007) 84–107
INFORMATION FUSION
www.elsevier.com/locate/inffus

An information fusion demonstrator for tactical intelligence processing in network-based defense ☆

Simon Ahlberg, Pontus Hörling, Katarina Johansson, Karsten Jöred, Hedvig Kjellström, Christian Mårtenson, Göran Neider, Johan Schubert, Pontus Svenson, Per Svensson *, Johan Walter

Department of Data and Information Fusion, Division of Command and Control Systems, Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden

Received 15 September 2004; received in revised form 16 November 2005; accepted 16 November 2005
Available online 20 January 2006

Abstract

The Swedish Defence Research Agency (FOI) has developed a concept demonstrator called the Information Fusion Demonstrator 2003 (IFD03) for demonstrating information fusion methodology suitable for a future Network Based Defense (NBD) C4ISR system. The focus of the demonstrator is on real-time tactical intelligence processing at the division level in a ground warfare scenario.

The demonstrator integrates novel force aggregation, particle filtering, and sensor allocation methods to create, dynamically update, and maintain components of a tactical situation picture. This is achieved by fusing physically modelled and numerically simulated sensor reports from several different sensor types with realistic a priori information sampled from both a high-resolution terrain model and an enemy organizational and behavioral model. This represents a key step toward the goal of creating in real time a dynamic, high fidelity representation of a moving battalion-sized organization, based on sensor data as well as a priori intelligence and terrain information, employing fusion, tracking, aggregation, and resource allocation methods all built on well-founded theories of uncertainty.

The motives behind this project, the fusion methods developed for the system, as well as its scenario model and simulator architecture are described. The main services of the demonstrator are discussed and early experience from using the system is shared.

Keywords: Scenario simulation; Demonstrator concept; Force aggregation; Terrain tracking; Sensor resource management; Sensor allocation; Sensor modelling; Terrain modelling

1. Introduction

In defense applications, information fusion processes exploit a dynamic target situation picture produced by multi-sensor fusion, combining its information with relevant a priori information, in order to refine and interpret a battlespace situation picture. Ultimately, this semi-automatic intelligence interpretation process aims at delivering a comprehensive picture of the opponents' options and, based on an evaluation of these options, suggest their likely intentions.

Along these lines, the Swedish Defence Research Agency (FOI) has developed a concept demonstrator called the Information Fusion Demonstrator 2003 (IFD03) for demonstrating in a tactical level ground warfare scenario, information fusion methodology expected to be suitable for use at the division or brigade level in a future Network-Based Defense (NBD) C4ISR system. Drawing upon progress reports presented in several conference papers [1–4], this article presents principles, methods, architecture, and conclusions from the development and early evaluation of

☆ A short version of this study was presented at the Seventh International Conference on Information Fusion (FUSION 2004) in Stockholm, Sweden [1].

* Corresponding author. Tel.: +46 8 5550 3696; fax: +46 8 5550 3700.
E-mail addresses: Per.Svensson@foi.se, pers@foi.se (P. Svensson).
URL: http://www.foi.se/fusion (P. Svensson).

1566-2535/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.inffus.2005.11.002

IFD03. We claim, and argue below, that our work represents a novel approach to providing an integrated simulation model of a complex system consisting of a battalion-sized moving ground force multi-target, being observed by a partly autonomous, partly controllable networked multi-sensor surveillance system and interpreted using information fusion. On the other hand, several other approaches to achieving improved situation awareness have been proposed and explored in the literature. Such related work is discussed in Section 5.

As detailed in this paper, IFD03 integrates methods related to different fusion "levels" [5], specifically *multi-sensor–multi-target tracking*, *force aggregation*, and *reactive multi-sensor management*. The information fusion methodology integrated into IFD03 rests on a few basic principles, i.e., cooperation between methods on fusion levels 1, 2, and 4, together with a tight coupling between a qualified synthetic environment and models of sensor behavior, target force behavior, and communication. It is based on Dempster–Shafer clustering and template matching [6–8], particle filtering [9,10], and finite set statistics [11].

In 2003, our project completed the development of IFD03 and performed a demonstration for an invited audience of tactical intelligence, C2 methodology, and information technology specialists.[1] The demonstration was based on a simple battalion-level ground force attack scenario (Section 1.4), originally formulated by the Swedish Armed Forces for studies in network-based defense.

1.1. Project background and rationale

On behalf of the Swedish Government, in 1998 the US-based defense consultancy company SAIC carried out a network-based defense architecture study entitled *Dominant Battlespace Awareness for the Swedish Armed Forces 2020*. The methodology of the study was based largely on scenario simulation. The most demanding, dimensioning scenario used was an attack on Swedish military forces by a division-sized ground force, supported by air and sea operations. An issue identified during the study as critically important was the need for a capability of the architecture to provide real-time information fusion while an intense flow of sensor and intelligence reports is being generated and communicated across the high-capacity network to the command posts. In the absence of purposeful information fusion, the anticipated information flow was perceived to become unmanageable. Thus the need for a focused research effort in information fusion was recognized by the Swedish Armed Forces.

The NATO Data Fusion Demonstrator (DFS) project, briefly discussed in [12] was at the time the best known of the few projects discussed in the open literature and in important ways represented the state-of-the-art. However, in relation to our requirements its architecture and methodology had severe shortcomings, see Section 1.2.

Preliminary studies of new methods of force aggregation based on Dempster–Shafer clustering and a literature survey of multi-sensor resource management were performed and reported by members of our group [13,14]. A lack of mathematically principled methods for multi-sensor, multi-target tracking of objects (usually vehicles) moving in terrain or other poorly characterized neighborhoods was already recognized in the information fusion community [15,16]. In this area, recent research progress in random set and particle filter methods seemed to us likely to yield useful results [11,16,17].

Partly based on insights gained in these studies, a research and development project was initiated at FOI whose task was to create, demonstrate and evaluate a "test-bed" to support development, demonstration, analysis, and evaluation of methods for fusing of surveillance and intelligence information and for coordination of surveillance resources.

The preliminary studies also showed that in order to build an information fusion test-bed or demonstrator, a number of stringent technical requirements had to be met by the simulation environment embedding the target, sensor, and fusion models. Key among these were [2,3]:

- a simulation platform tightly coupled to the "fusion engine", enabling the representation of closed-loop management of sensor resources,
- terrain modelling functionality capable of representing a large geographical area in high spatial resolution and three dimensions, as well as of performing state-of-the-art line-of-sight and trafficability calculations in partly covered terrain,
- a simulation engine capable of efficiently representing and dynamically managing a battalion-sized moving ground force, as well as several sensor platforms concurrently,
- tools for scenario definition usable by non-programmers.

1.2. Critical issues

Kent [12] provides a critical review of information fusion as a technology for commanders of land-based military operations, while noting that there has been relatively little open research in this area, an observation which is still largely valid. In particular, it cites experience from the NATO Data Fusion Demonstrator (DFS) project, finished in 1998, which among other things concluded that future research should consider from the outset information display methods and information manipulation by humans. On the other hand, NATO DFS did produce some critical intelligence information more accurately and timely than the conventional manual process. The paper's general assessment is, however, that "fusion research has failed

[1] A simplified version of the demonstration is available at http://www.foi.se/fusion/avi/IFD03_demo.avi.

to produce systems that can support the needs of the commanders of land-based military operations, especially those that involve operations other than war''.

Other researchers, including Lorenz and Biermann [18] and Lambert [19], make similar remarks based on first-hand experience with prototype or demonstrator systems while emphasizing the fact that after the end of the cold war, in most western countries the focus of interest has shifted from large-scale ''symmetric'' military battles to asymmetric peace-keeping and peace-enforcement coalition operations. Lorenz and Biermann [18] claim that in today's conflicts, ''the major problem [in meeting this] new challenge is the lack of ad-hoc availability of necessary background knowledge for automated analysis tasks. [...] We are forced to match every change according to a collection of standard patterns and develop a catalogue of recurring changes to patterns.''

Since NATO DFS was not intended as a fieldable system but only as a demonstrator, we believe that this criticism is somewhat premature; time was not then, and is still not ripe for a fully mature system for tactical information fusion to be built, in particular not one suitable for operations other than war. As indicated in Section 1.1, however, in relation to our requirements the methodology and architecture of NATO DFS had other severe shortcomings:

- its fusion and aggregation methods were not based on mathematical, nor even quantitative, theories of uncertainty,
- as a consequence, it was impossible to provide mathematical estimates of the credibility of the system's conclusions, e.g., with regard to report association and force aggregation,
- it did not provide any theoretically well-founded method for multi-target tracking of mobile vehicles in terrain,
- since it lacked capability to track and predict the position of adversary targets, it could not be used for modelling closed-loop automated management of the multi-sensor resource,
- its simulation and terrain-modelling architecture was unable to support the required integrated, high-fidelity representation of the intelligent multi-target, multi-sensor, terrain-interacting, communicating, mobile system to be modelled.

In the remainder of this paper, the techniques developed for IFD03 to resolve these issues are discussed.

1.3. Conceptual overview

In general terms, the scenario simulation in IFD03 describes the stochastic interaction between an observation system, a complex target system, in this case a hierarchically organized adversary unit, and a complex environment. Our long-term objective has been to create a dynamic, high fidelity representation of the behavior of a moving battalion-sized organization, based on sensor data as well as a priori intelligence and terrain information. Such a model would allow accurate short-term predictions to be made of adversary movement, tactical disposition and capability, providing an opportunity for computer-supported real-time tactical planning of countermeasures.

Surveillance information from the observation system is generated during the simulation by a set of sensor models. The sensors deliver reports more or less continuously to a fusion node, symbolizing a future division-level intelligence staff. In the fusion node information is fused, interpreted and fed back to sensors as control messages in simulated real time. The fusion node uses the sensor information as input to aggregation, tracking, and reactive sensor management processes (see Section 2).

The fusion node communicates with the reconnaissance resources of the observation system, i.e., a civilian, temporarily connected roadside video camera, Home Guard soldiers using advanced position-measuring binoculars, ground sensor networks detecting and classifying moving vehicles, and surveillance UAVs remotely controlled by the fusion node and carrying video or IR cameras. Communications intelligence (COMINT) surveillance units are also available. The UAVs normally fly along predetermined routes, while being capable of immediately obeying control messages from the fusion node, such as to switch to a different route or to deploy a ground sensor network.

The target system consists of ''red'' (adversary) forces of battalion strength, with several mechanized and armored subunits. These units move largely according to doctrinal rules on or near roads. Their speed and movement pattern is influenced by road and terrain trafficability according to unit and vehicle type.

1.4. Scenario

The scenario takes place in May 2015. Tension in the Baltic Sea area has grown gradually over several years. At the outbreak of the war, a ''trojan horse'' landing at the Kapellskär ferry harbor, 90 km north-east of Stockholm, is judged to constitute the greatest threat.

The only intelligence sources available at the time of the landing are four Home Guard patrols deployed at strategic points along the enemy advance routes, Fig. 1. The battalion's UAV group is ordered to immediately deploy two UAVs for reconnaissance.

Forty-five minutes later, the two UAVs directed to Rådmansö have contributed to creating a fairly detailed situation picture. The chief intelligence officer is now able to state that the adversary force consists of a mechanized battalion reinforced by antiaircraft and mortar units, advancing along two roads towards the town of Norrtälje.

The final phase of the scenario involves the continued adversary march towards the tactically critical lake passes south-west of the town. As sensor platforms become fewer and eventually only a single UAV remains, that resource needs to be intelligently utilized to estimate which routes

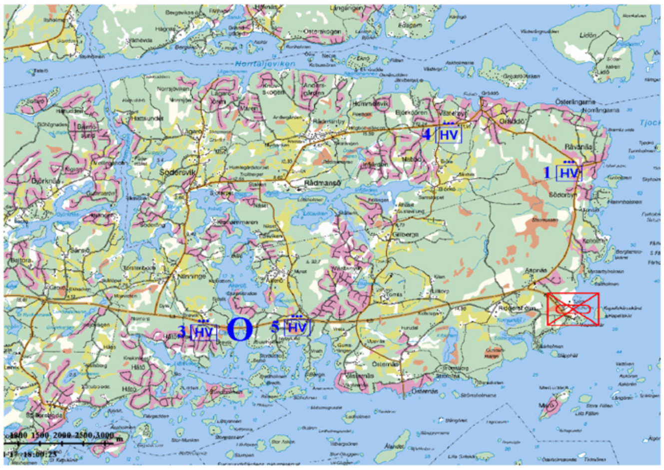

Fig. 1. Information collection situation at 17.00. Four Home Guard (HV) patrols are located at critical points along the adversary's approach route. A bridge is located at O.

the adversary units are likely to take, and when they will reach the lake passes. Hence, the automatic sensor resource manager of the fusion node is tasked to find the best route for the UAV and to decide where to drop its deployable ground sensor network.

1.5. Use cases

The major use cases [20] we had in mind when creating the system were:

- performing demonstrations addressing a possibly "information-fusion-naive" audience. This is communication, not research, but could be developed into a methodology to present, visualize, and later analyze in detail properties of new components and subsystems,
- performing studies and experiments with sensor models, terrain and other environment models, fusion methods, doctrine models, scenario assumptions, etc., in various combinations, to test different hypotheses about possibilities and limitations related to NBD and information fusion,
- developing methodology and models for information fusion, i.e., specification, development, and testing of new methods and fusion concepts.

1.6. Development tools

Whereas previous information fusion projects at FOI have focused on method and algorithm development for various specific problems, in particular clustering, aggregation, and classification of force units [13] and sensor management [14], it was realized from the start of the IFD03 project [2] that its functionality requirements would compel us to devote significant resources to acquire and develop a set of new and complex system development tools. Thus, the development tools used in the project were chosen to support substantial reuse [21], including evolutionary extension and rewriting [22], both of software models and algorithms and of simulation scenario scripts.

Consequently, the demonstrator implementation is built on top of three comprehensive commercial development environments, the *problem solving environment* (for an in-depth study of this concept, see [23]) MATLAB™ [24], the *simulation framework* FLAMES™ [25], and the *terrain modelling system* TerraVista™ Pro Builder [26]. In the project, FLAMES and MATLAB were tightly integrated, and FLAMES' new handling of advanced terrain models, generated by TerraVista, was specified and at least partly financed. Finally, the FLAMES software for visualization of simulation results using the new terrain modelling feature was restructured and both functionally and computationally substantially improved.

1.7. Structure of paper

Section 2 reviews the fusion methods used in the demonstrator. These are Dempster–Shafer clustering and Dempster–Shafer template matching for force aggregation (Section 2.1), probability hypothesis density (PHD) particle filtering for ground vehicle tracking (Section 2.2), and random set simulation for sensor allocation (Section 2.3).

In Section 3 the software architecture of the IFD03 system is presented. Section 3.1 introduces the main object categories and their roles in the simulation. Principles and design requirements of object and doctrine models are surveyed in Section 3.2. In Section 3.3, a viewer's perspective of the demonstration is first introduced, then in Section 3.3.1, the organization of the visualization module in IFD03 is described. Modeling techniques used in creating the environment model of the demonstrator are discussed in Section 3.4.

Section 4 describes a set of qualitative analyses of IFD03 output carried out to evaluate the performance of its fusion methods. Section 5 provides a comparison with other published approaches to building information fusion demonstrator and prototype systems. Section 6 concludes the paper.

2. Fusion methods

The analysis module has three main tasks and uses four different methods. The tasks are force aggregation, ground vehicle tracking and sensor allocation. They are performed using Dempster–Shafer clustering and template matching for force aggregation, probability hypothesis density (PHD) particle filtering for ground vehicle tracking, and random set simulation for sensor allocation.

2.1. Force aggregation

In force aggregation, sensor reports with given position, time, and type information are used. Here, force aggregation is defined as a sequence of two processes: (1) association of intelligence reports, objects or units (depending on hierarchical level) by a clustering process; (2) classification of cluster content through comparison with templates.

Initially, all pairs of intelligence reports are evaluated, to find whatever is against an association of these two reports to the same object: Wrong type of vehicle? (note that type assignments are allowed to be more or less specific) Is distance too long or too short? Wrong direction? Wrong relative positions? etc. This yields a conflict matrix which is supplied to the clustering algorithm. We use the Dempster–Shafer clustering algorithm [27,28,7,8] to partition the set of reports into subsets corresponding to objects, and classify the objects by fusing all intelligence using Dempster's rule. This method continues upwards level by level. At the vehicle to platoon level, vehicles are clustered and groups of vehicles are classified using Dempster–Shafer matching against templates [6]. At all levels in clustering and template matching several alternative hypotheses are carried. Each alternative hypothesis is matched and evaluated against all templates and a weighted average of fitness is calculated for each potential template.

Screen pictures from the demonstrator showing the result of automated force aggregation at the platoon and company levels are shown in Figs. 2 and 3. A few other approaches to force aggregation are [29,18,30].

2.1.1. Conflict matrix

There is one conflict matrix for each aggregation level. The conflict matrix element c_{ij} contains the conflict between the entities i and j. The matrix is symmetric and contains zeros on the diagonal.

When computing the conflict matrix for the reports, the conflict between two reports is based on their vehicle type, on how fast a vehicle must travel in order to cause the two reports and on how much their directions differ. When computing the conflict matrix for vehicles and units, the conflicts are based on doctrine data that specify how far apart the objects appear within their unit.

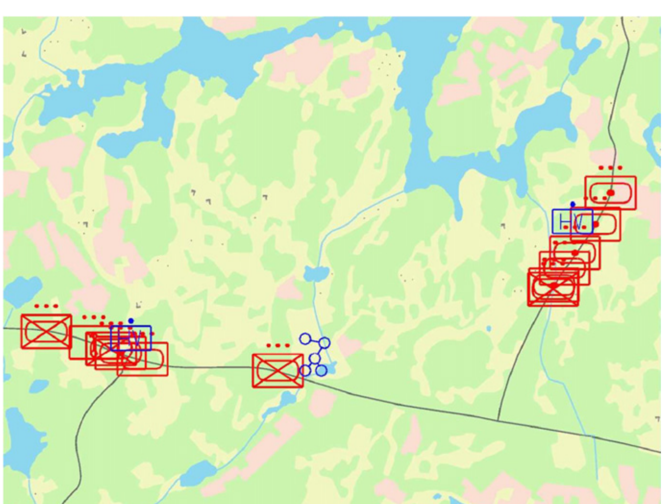

Fig. 2. Force aggregation of vehicles into platoons.

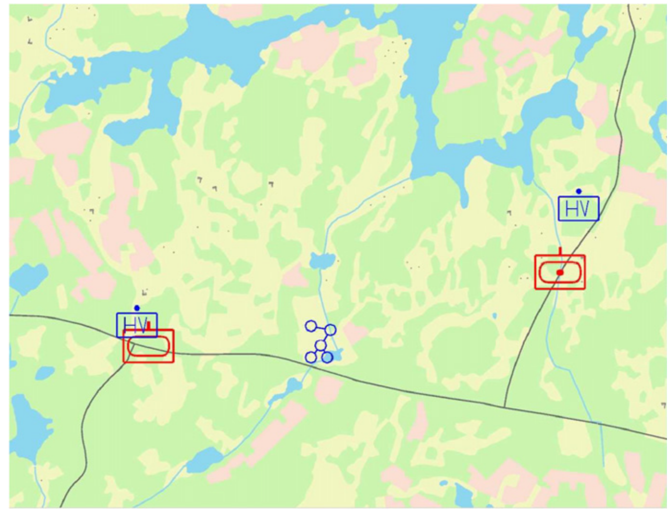

Fig. 3. Force aggregation of platoons into companies.

All entities—reports, vehicles and units—contain a classification of types, TY. However, the classification is uncertain, so we can only give probabilities for sets of types, representing the varying specificity of type assignments in the reports. The basic belief mass supporting that an entity is of type $A \in TY$ is denoted $m(A)$. All basic belief functions in IFD03 are consonant, i.e., the focal elements of the belief functions can be ordered by set inclusion, ensuring that their type conflicts are well-defined.

Conflict matrix for reports. The value c of an element in the conflict matrix is computed from the type conflict c^{t}, the speed conflict c^{s} and the direction conflict c^{d}.

$$c = 1 - (1 - c^{\mathrm{t}})(1 - c^{\mathrm{s}})(1 - c^{\mathrm{d}}) \tag{1}$$

The type conflict between the entities e_i and e_j is given by Dempster's rule of combination [31,32]:

$$c_{ij}^{\mathrm{t}} = \sum_{\substack{A \in e_i, B \in e_j \\ A \cap B = \emptyset}} m(A) \cdot m(B) \tag{2}$$

The speed conflict is obtained by calculating the speed at which a vehicle must travel in order to cause both reports, see Fig. 4.

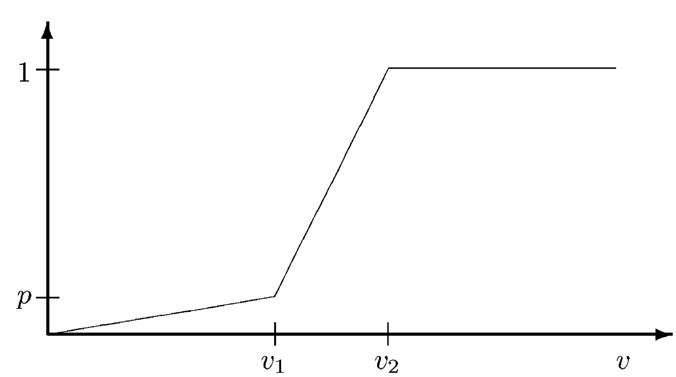

Fig. 4. The relationship between speed and conflict.

The direction conflict is calculated in an analogous way by computing the difference between the directions of movement in the two reports. For details see [13].

Conflict matrix for vehicles and units. If there is no enemy unit that contains the estimated entity types of both reports, the conflict should be large. If there is such a unit, the conflict is determined by the current estimate of distance between the units and the maximum allowed distance according to doctrine.

For each level—vehicles, platoons, companies, etc.—a distance matrix, DM, is defined so that the allowed distance between TY_a and TY_b is DM_{ab}.

The conflict between entity e_i and e_j is given by

$$c_{ij} = \sum_{A \in e_i, B \in e_j} Q_{AB} \cdot m(A) \cdot m(B) \tag{3}$$

where

$$Q_{AB} = \min_{a \in A, b \in B} q_{ab} \tag{4}$$

and

$$q_{ab} = \begin{cases} 1 & d > DM_{ab} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where d is the distance between entities e_i and e_j.

2.1.2. Clustering

In [28] a method for clustering intelligence reports based on their pairwise conflict was developed. This method was extended into a method capable of handling also pairwise attractions [7]. Such evidence is not generated intrinsically in the same way as conflicts. Instead, it is provided by communications intelligence, indicating that two objects probably belong to the same unit (cluster) as they are in communication. Such information is made available from studying communication patterns obtained through COMINT, e.g., if two objects are transmitting in sequence one may calculate a probability that they are in communication and thus belong to the same unit structure.

As conflicts push reports apart (into different clusters) and attractions pull them together (into the same cluster), using both can lead to an improved clustering result and faster computation. The conflict, as calculated in Section 2.1.1, and the attraction together form the basis for separating intelligence reports into clusters. A high conflict between two intelligence reports is an indication of repellency that they do not belong to the same cluster. The higher the conflict, the less credible it is that they belong to the same cluster.

Attracting evidence is represented as a pairwise piece of evidence, where p_{ij} is a degree of attraction.

The best partitioning of all intelligence reports is found by a clustering process [8] which minimizes a function $m_{\{\chi_a\}\oplus\chi}(\neg AdP)$ with a proposition that this is not an "adequate partition" AdP. This function was derived by combining the conflicting information from the clusters $m_{\{\chi_a\}}$ with the attracting metalevel evidence m_χ. For details, see [7]. Here, we use the same notation as there.

Approximately this function can be written as

$$m_{\{\chi_a\}\oplus\chi}(\neg AdP) \approx 1 - \prod_{(ij)|\forall a.e_i \wedge e_j \notin \chi_a} (1 - p_{ij}) \times \prod_a \times \prod_{(ij)|e_i \wedge e_j \in \chi_a} (1 - c_{ij}) \tag{6}$$

Clustering of the intelligence reports is done by neural clustering using Potts spin theory [33,34]. The Potts spin problem consists of minimizing an energy function

$$E = \frac{1}{2} \sum_{i,j=1}^{N} \sum_{a=1}^{q} (J_{ij}^- - J_{ij}^+) S_{ia} S_{ja} \tag{7}$$

by changing the states of the spins S_{ia}, where $S_{ia} \in \{0,1\}$ and $S_{ia} = 1$ means that report i is in cluster a. N is the number of intelligence reports and q the number of clusters. This model serves as a clustering method if J_{ij}^- is used as a penalty factor when reports i and j are in the same cluster, and J_{ij}^+ when they are in different clusters.

The minimization is carried out by deterministic annealing [35]. For computational reasons a mean field model is used, with $V_{ia} = \langle S_{ia} \rangle$, $V_{ia} \in [0,1]$, in order to find the minimum of the energy function. The Potts mean field equations are

$$V_{ia} = \frac{e^{-H_{ia}[V]/T}}{\sum_{b=1}^{q} e^{-H_{ib}[V]/T}} \tag{8}$$

where

$$H_{ia}[V] = \sum_{j=1}^{N} J_{ij} V_{ja} - \gamma V_{ia}, \tag{9}$$

and V, T, H_{ib} and γ are parameters of the annealing process.

In order to map the function $m_{\{\chi_a\}\oplus\chi}(\neg AdP)$ onto a Potts spin neural network it must be rewritten as a sum of terms.

Minimizing the right member of Eq. (6) is equivalent to minimizing the expression:

$$\sum_{(ij)|\forall a.e_i \wedge e_j \notin \chi_a} -\log(1 - p_{ij}) + \sum_a \sum_{(ij)|e_i \wedge e_j \in \chi_a} -\log(1 - c_{ij}) \tag{10}$$

To apply the Potts model to Dempster–Shafer clustering, interactions $J_{ij}^- = -\log(1 - c_{ij})\delta_{|A_i \cap A_j|}$ and $J_{ij}^+ = -\log(1 - p_{ij})(1 - \delta_{|A_i \cap A_j|})$ are used in the energy function (Eq. (7)), where A_i is the focal element of the simple support function e_i, and

$$\delta_{|A_i \cap A_j|} \equiv \begin{cases} 1 & |A_i \cap A_j| = 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

and δ_{ij} (in Fig. 5) is a Kronecker function

$$\delta_{ij} \equiv \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{12}$$

INITIALIZE

K (number of clusters); N (number of intelligence reports)

$J^-_{ij} = -log(1 - c_{ij})\delta_{|A_i \cap A_j|} \forall i, j;$

$J^+_{ij} = -log(1 - p_{ij})(1 - \delta_{|A_i \cap A_j|}) \forall i, j;$

$s = 0; t = 0; \epsilon = 0.001; \tau = 0.9; \gamma = 0.5;$

$T^0 = T_c$ (a critical temperature) $= \frac{1}{K} \cdot max(-\lambda_{min}, \lambda_{max})$,

where λ_{min} and λ_{max} are the extreme eigenvalues of M,

where $M_{ij} = J^-_{ij} - J^+_{ij} - \gamma\delta_{ij};$

$V^0_{ia} = \frac{1}{K} + \epsilon \cdot rand[0, 1] \forall i, a;$

REPEAT

REPEAT-2

$\forall i$ Do:

$H^s_{ia} = \sum_{j=1}^{N} (J^-_{ij} - J^+_{ij}) V_{ja}^{\left\{\begin{smallmatrix} s+1 & j<i \\ s & j\geq i \end{smallmatrix}\right.} - \gamma V^s_{ia} \forall a;$

$F^s_i = \sum_{a=1}^{K} e^{-H^s_{ia}/T^t};$

$V^{s+1}_{ia} = \frac{e^{-H^s_{ia}/T^t}}{F^s_i} + \epsilon \cdot rand[0, 1] \forall a;$

$s = s + 1;$

UNTIL-2

$\frac{1}{N} \sum_{i,a} |V^s_{ia} - V^{s-1}_{ia}| \leq 0.01;$

$T^{t+1} = \tau \cdot T^t;$

$t = t + 1;$

UNTIL

$\frac{1}{N} \sum_{i,a} (V^s_{ia})^2 \geq 0.99;$

RETURN

$\{\chi_a | \forall S_i \in \chi_a, \forall b \neq a V^s_{ia} > V^s_{ib}\};$

Fig. 5. Pseudocode for clustering algorithm.

In Fig. 5 an algorithm for minimizing the energy function through iteration of Eqs. (8) and (9) is shown.

2.1.3. Number of clusters

In order to estimate the correct number of clusters, the conflict function which results from clustering with different numbers of clusters is calculated and analyzed. Of course, where the number of clusters is too small the conflict will be high. Where the number of clusters is too large a small residual conflict, emanating from measurement errors, will remain.

It was found experimentally that there is a change of behavior in the conflict function near the correct number of clusters, which was determined as follows.

1. The logarithm of the total weight of conflict as function of the number of clusters was computed from empirical data, see Fig. 6.
2. The concave lower envelope of this function was determined using a convex hull algorithm.
3. At an arbitrary abscissa, the envelope function was bisected in a left and a right part, each of which were then fitted by least squares to a straight line.
4. The acute angle between the two lines was maximized over all bisection abscissas and the maximizing abscissa was chosen as number of clusters.

This is similar to the L-method by Salvador and Chan [36].

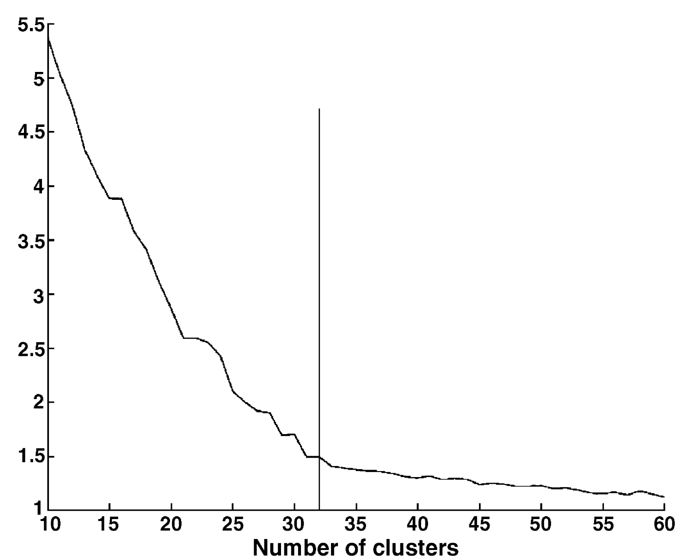

Fig. 6. Logarithm of the total weight of conflict, showing a qualitative change in behavior near the correct number of clusters. The result in this figure is based on clustering all intelligence obtained from sensors observing the advance of one mechanized battalion.

Experimental tests of this algorithm using sensor reports and vehicles from the scenario described in Section 1.4 showed good correspondence between the number of clusters determined in this way and the total number of observed vehicles.

2.1.4. Classification

The classification process deals with intelligence reports on a cluster-by-cluster basis. Looking at intelligence in one of the clusters, the classification from intelligence using templates takes place in two phases. First, all intelligence reports within the cluster are combined, then the combined intelligence is compared with all available templates.

In the combination of intelligence a special concern is the representation used. As the reports in general are not reports about the same object or group of objects, one cannot use a simple representation dealing only with object type.

Instead, a more complex representation has to be used, that allows keeping track of different objects and their possible types. Intelligence reports that are judged to be referring to the same object or group of objects are precombined and henceforth viewed as one intelligence report. In this way, all intelligence reports in the cluster under investigation can be combined, providing the opportunity to investigate different resulting hypotheses regarding force composition.

When selecting a template for the current cluster, a best match between template and fused intelligence is sought. Since intelligence consists of multiple alternative hypotheses with an accompanying uncertainty, every hypothesis, to its degree of uncertainty, must be taken into account. As these hypotheses are also non-specific with regard to object type, i.e., they refer to a subset of all possible types instead of to a single type, one cannot expect a perfect match for each type of object in the template. Instead, one should look for the possibility of a match between intelligence and template, i.e., the absence of conflicts in number of items between what the intelligence proposes and what each available template requests for all subsets of types. With this measure a template can be selected for intelligence with non-specific propositions.

Let us now focus on one subset χ_a and the aggregation of the intelligence in this subset. Let $TY = \{TY_i\}$ be a set of all possible types of objects, where TY_i is a type of vehicle or a type of unit, depending on which hierarchical level is analyzed.

When fusing reports regarding different sets of objects that should be combined as components of a larger unit structure, the frame of discernment becomes

$$\Theta_{I_a} = \{\langle x_1, x_2, \ldots, x_{|I_a|}\rangle\} \tag{13}$$

where $x_i = (x_{i\bullet}n, x_{i\bullet}pt)$ is information regarding the ith set of objects with $x_{i\bullet}n \subseteq \{1, \ldots, N_{C_a}\}$ and $x_{i\bullet}pt \subseteq TY$. Here, N_{C_a} is the maximum number of objects according to the intelligence in cluster a.

One now needs to compare templates having specific propositions that are certain in what they are requesting with intelligence propositions that are not only uncertain but may also be non-specific in what they are supporting. This problem is handled by comparing a candidate template and intelligence with each subset of TY. To do this, one may investigate how much support a subset of TY receives both directly and indirectly from intelligence and template, respectively.

The support for a subset of TY is added up from all propositions that are equal to, or itself a subset of this subset of TY. This is similar to the calculation of belief from basic probability numbers in Dempster–Shafer theory, except that one does not add basic probability numbers but natural numbers representing the number of objects of the proposed types.

Let $TE = \{TE_i\}$ be a set of all available templates. Each template is represented by any number of slots S_i^j where $S_{i\bullet}^j pt \in TY$ is a possible type and $S_{i\bullet}^j n$ is the number of that type in TE_i.

Since there are several different alternative propositions in the intelligence regarding the type of objects and their corresponding number of objects, one needs to compare each potential template with these alternatives and let each proposition influence the evaluation. For each template a measure of fitness is found between the template and each proposition in the intelligence separately.

A linear combination is then made, where each measure of fitness is weighted by the basic probability number of that proposition,

$$m_{\oplus J_a}(\langle x_1, x_2, \ldots, x_{|I_a|}\rangle), \tag{14}$$

giving

$$\begin{aligned}
\pi_{\oplus J_a}(TE_i) = \\
&\frac{1}{2} \sum_{\langle x_1, x_2, \ldots, x_{|I_a|}\rangle} m_{\oplus J_a}(\langle x_1, x_2, \ldots, x_{|I_a|}\rangle) \\
&\times \left[\max_{n \in SC_a(TY)} \left\{ \min \left[\frac{n}{ST_i(TY)}, \frac{ST_i(TY)}{n} \right] \right\} \right. \\
&\left. + \min_j \left[\left\{ \begin{array}{r} \max_{n \in SC_a(S_{a\bullet}^j pt)} \\ \left\{ \min \left[\frac{n}{ST_i(S_{a\bullet}^j pt)}, \frac{ST_i(S_{a\bullet}^j pt)}{n} \right] \right\}, \\ ST_i(S_{a\bullet}^j pt) > 0 1, \\ ST_i(S_{a\bullet}^j pt) = 0 \end{array} \right\} \right] \right]
\end{aligned} \tag{15}$$

where $S_{a\bullet}^j pt \subseteq TY$.

These functions were derived in [6]. Here we use essentially the same notation as there.

For each TE_i, the number of objects requested by the template in Eq. (15) is calculated as

$$ST_i(X) = \sum_{j | S_{i\bullet}^j pt \subseteq X_\bullet pt} S_{i\bullet}^j n \quad \forall X \subseteq TY \tag{16}$$

and the number of objects supported by proposition $\langle x_1, x_2, \ldots, x_{|I_a|}\rangle$ of the intelligence as

$$\begin{aligned}
&SC_a(X | \langle x_1, x_2, \ldots, x_{|I_a|}\rangle) \\
&\quad = \sum_{i \,|\, x_i \in \langle x_1, x_2, \ldots, x_{|I_a|}\rangle x_{i\bullet}pt \subseteq X_\bullet pt} x_{i\bullet}n \quad \forall X \subseteq TY
\end{aligned} \tag{17}$$

To summarize, the available intelligence is first fused into several alternative hypotheses. Each hypothesis is then evaluated against all templates to give an overall fitness for each template. Finally, the best fitting template is selected if its fitness value is above a predefined threshold. While the fitness measure $\pi_{\oplus J_a}(\cdot)$ is used for aggregation from the current hierarchical level, one also needs the basic probability of the highest ranked template for any further aggregation from the next hierarchical level. Through a fitness weighted transformation, these templates will share this support in relation to their fitness towards the corresponding focal element in the intelligence.

The basic probability number of TE_i is found as

$$m_{\oplus J_a}(TE_i) = \sum_{\langle x_1,x_2,\ldots,x_{|I_a|}\rangle \supseteq TE_i} \left[m_{\oplus J_a}(\langle x_1,x_2,\ldots,x_{|I_a|}\rangle) \times \frac{\pi_{\langle x_1,x_2,\ldots,x_{|I_a|}\rangle}(TE_i)}{\sum_{TE_j \subseteq \langle x_1,x_2,\ldots,x_{|I_a|}\rangle} \pi_{\langle x_1,x_2,\ldots,x_{|I_a|}\rangle}(TE_j)} \right] \tag{18}$$

The evidential force aggregation method makes it possible to aggregate uncertain intelligence reports with multiple uncertain and non-specific propositions into recognized forces using templates.

2.2. Tracking

In the tracking module, the states of an unknown number of ground vehicles moving in terrain are maintained. The tracking is based on observations in the form of intelligence reports of ground position $\mathbf{y}$, ground speed v, and direction of motion θ.

When tracking multiple targets in general, the size of the state-space for the joint distribution over target states grows exponentially with the number of targets. When the number of targets is large, this makes it impossible in practice to maintain the joint distribution over target states.

A mathematically principled approach to avoid the combinatorial explosion is to propagate only the first moment of the joint distribution, the *probability hypothesis density* (*PHD*) [17]. This entity is briefly described in Section 2.2.1. It has the property that its integral over each sub-area S in the state-space is the expected number of targets within this area. Peaks in the PHD can thus be regarded as estimated target states. Since the identities of objects are not maintained, there is no model-data association problem. However, the method has the drawback that no knowledge about dependencies in motion between objects can be represented. Also in Section 2.2.1, a particle filter [9,37] implementation of PHD tracking, the *PHD particle filter*, is briefly described. For a thorough description, see [38]. Particle filtering is suited for tracking with non-linear and non-Gaussian motion models, and is thus suitable for ground target tracking. The non-linear terrain-dependent motion model is described in Section 2.2.2.

In [38], the sensor visibility is assumed constant with respect to position and time. Here, we incorporate knowledge of sensor quality and field of view into the filter. This is described in Section 2.2.3.

2.2.1. PHD filtering

The number of vehicles (called targets below) to track is unknown and varies over time. This means that the targets at time t is a *random set* [39,11] $\Gamma_t = \{\mathbf{X}_t^1, \ldots, \mathbf{X}_t^{N_t}\}$, where $\mathbf{X}_t^i$ is the state vector of target i and N_t is the number of targets in the set. A certain outcome of the random set Γ_t is denoted $X_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t}\}$. Similarly, the set of observations received at time t is a random set $\Sigma_t = \{\mathbf{Z}_t^1, \ldots, \mathbf{Z}_t^{M_t}\}$, where M_t can be larger than, the same as, or smaller than N_t. A certain outcome of the random set Σ_t is denoted $Z_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^{m_t}\}$.

For large numbers of targets, it is computationally intractable to keep track of every single target. A more tractable approach is then to represent the first moment of the full joint distribution, the probability hypothesis density (PHD) $D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t})$ [17,38], which is defined over the state-space Θ of one target instead of the much larger joint target space Θ^{N_t}. The computational cost of propagating the PHD over time is much lower than propagating the full distribution.

The PHD has the properties that, for any subset $S \subseteq \Theta$, the integral of the PHD over S is the expected number of targets in S at time t:

$$E[|\Gamma_t \cap S|] = \int_S D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t})\mathrm{d}\mathbf{x}_t \tag{19}$$

In other words, it will have local maxima approximately at the target locations. The integral of the PHD over Θ is the expected number of targets, n_t.

We now describe one time-step in the PHD filter, which is propagated using Bayes' rule [17,38]. First, a *prior* PHD is predicted from the PHD and observations at the previous time-step. Then, new observations are used to compute the *likelihood* of this prior PHD. This results in a new *posterior* PHD. The steps are described below.

Prediction. The temporal model of the targets include birth (appearance of a target in the field of view), death (disappearance of a target from the field of view) and temporal propagation. Probability of target death is p_D and of target birth p_B.

Target hypotheses are propagated from earlier hypotheses according to the dynamical model

$$\mathbf{X}_t = \phi(\mathbf{X}_{t-1}, \mathbf{W}_t) \tag{20}$$

where $\mathbf{W}_t$ is a noise term independent of $\mathbf{X}_{t-1}$ (Section 2.2.2). This gives

$$f_{\mathbf{X}_t|\mathbf{X}_{t-1},\mathbf{Z}_{1:t-1}}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{z}_{1:t-1}) \equiv f_{\mathbf{X}_t|\mathbf{X}_{t-1}}(\mathbf{x}_t|\mathbf{x}_{t-1})$$

with no dependence on the history of observations $\mathbf{z}_{1:t-1}$.

Other target hypotheses are born from observations at the previous time instant [38] according to the model

$$\mathbf{X}_t = \phi(h_{\mathbf{X}_t}^{-1}(\mathbf{Z}_{t-1}, \mathbf{V}_{t-1}), \mathbf{W}_t) \tag{21}$$

where $\mathbf{V}_t$ is a noise term (Section 2.2.2). This model defines the birth pdf $f_{\mathbf{X}_t|\mathbf{Z}_{t-1}}(\mathbf{x}_t|\mathbf{z}_{t-1})$.

To take all observations $\Sigma_t = \{\mathbf{Z}_t^1, \ldots, \mathbf{Z}_t^{M_t}\}$ into account for target birth, a birth PHD is defined from the set of birth pdfs as

$$D_{\mathbf{X}_t|\Sigma_{t-1}}(\mathbf{x}_t|Z_{t-1}) = \sum_{\mathbf{z}_{t-1}^i \in Z_{t-1}} f_{\mathbf{X}_t|\mathbf{Z}_{t-1}}(\mathbf{x}_t|\mathbf{z}_{t-1}^i). \tag{22}$$

Given the models of motion, death and birth, the prior PHD [17] is estimated from the posterior PHD at the previous time instant as

$$\begin{aligned} &D_{\mathbf{X}_t|\Sigma_{1:t-1}}(\mathbf{x}_t|Z_{1:t-1}) \\ &\quad = p_B D_{\mathbf{X}_t|\Sigma_{t-1}}(\mathbf{x}_t|Z_{t-1}) \\ &\qquad + \int (1-p_D) f_{\mathbf{X}_t|\mathbf{X}_{t-1}}(\mathbf{x}_t|\mathbf{x}_{t-1}) D_{\mathbf{X}_{t-1}|\Sigma_{1:t-1}}(\mathbf{x}_{t-1}|Z_{1:t-1})\mathrm{d}\mathbf{x}_{t-1} \end{aligned} \tag{23}$$

Observation. We define p_{FN} as the probability that a target is *not* observed at a given time step (the probability of false negative). This entity is further discussed in Section 2.2.3. Assuming that there are no spurious observations, the posterior PHD distribution is computed [17] from the prior as

$$\begin{aligned} D_{\mathbf{X}_t|\Sigma_{1:t}}(\mathbf{x}_t|Z_{1:t}) = &\sum_{\mathbf{z}_t^i \in Z_t} f_{\mathbf{X}_t|\mathbf{Z}_t,\Sigma_{1:t-1}}(\mathbf{x}_t|\mathbf{z}_t^i, Z_{1:t-1}) \\ &+ p_{FN} D_{\mathbf{X}_t|\Sigma_{1:t-1}}(\mathbf{x}_t|Z_{1:t-1}) \end{aligned} \tag{24}$$

where

$$f_{\mathbf{X}_t|\mathbf{Z}_t,\Sigma_{1:t-1}}(\mathbf{x}_t|\mathbf{z}_t^i, Z_{1:t-1}) \propto f_{\mathbf{Z}_t|\mathbf{X}_t}(\mathbf{z}_t^i|\mathbf{x}_t) D_{\mathbf{X}_t|\Sigma_{1:t-1}}(\mathbf{x}_t|Z_{1:t-1}) \tag{25}$$

which is a pdf (with the integral 1 over the state-space).

Using Eqs. (22)–(24), the PHD is propagated in time. Pseudocode for one time step of the tracking algorithm is shown in Fig. 7. The result of the tracking is the estimated number of targets and the location of the detected maxima in the posterior PHD in each time step. An example of a posterior PHD is shown in Fig. 8.

```
% prediction:
for j ← 1 : n_{t-1}N
    % from previous time-step:  posterior ξ^j_{t-1}.
    sample prior ξ̃^j_t from f_{X_t|X_{t-1}}(x_t | ξ^j_{t-1}).
    compute prior weight ϖ^j_t ← (1 − p_D)/N.
end for
for i ← 1 : m_{t-1}
    % from previous time-step:  observation z^i_{t-1}.
    J ← j.
    for j ← (J + 1) : (J + N)
        sample prior ξ̃^j_t from f_{X_t|Z_{t-1}}(x_t | z^i_{t-1}).
        compute prior weight ϖ^j_t ← p_B/N.
    end for
end for

% observation:
for j ← 1 : (m_{t-1} + n_{t-1})N
    compute likelihood π^{0,j}_t ← p_FN ϖ^j_t.
end for
for i ← 1 : m_t
    for j ← 1 : (m_{t-1} + n_{t-1})N
        compute likelihood π̃^{i,j}_t ← ϖ^j_t f_{Z_t|X_t}(z^i_t | ξ̃^j_t).
    end for
    for j ← 1 : (m_{t-1} + n_{t-1})N
        normalize likelihood π^{i,j}_t ← π̃^{i,j}_t / Σ_k π̃^{i,k}_t.
    end for
end for

% resampling:
expected number of targets n_t = Σ_{i=0}^{m_t} Σ_{j=1}^{(m_{t-1}+n_{t-1})N} π^{i,j}_t.
for j ← 1 : n_t N
    monte carlo sample posterior ξ^j_t from weighted set
    ∪_{i=0}^{m_t} {(ξ̃^1_t, π^{i,1}_t), ..., (ξ̃^{(m_{t-1}+n_{t-1})N}_t, π^{i,(m_{t-1}+n_{t-1})N}_t)}.
end for
```

Fig. 7. Pseudocode for time-step t in a PHD particle filter.

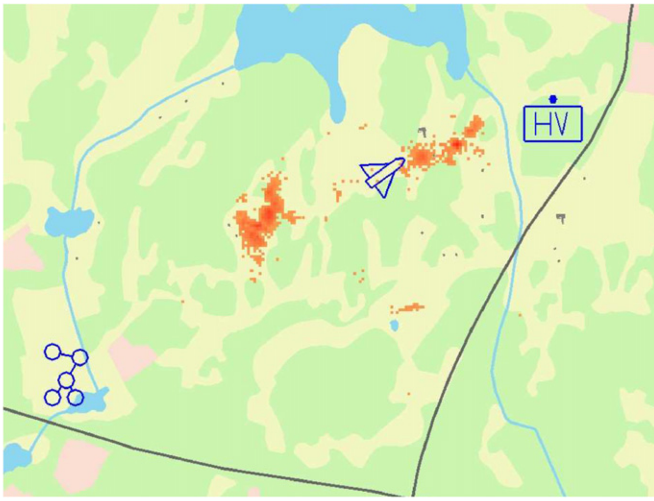

Fig. 8. The posterior PHD represented as a set of particles. For greater visibility, the histogram over particle position is shown; the saturation of red in a certain sub-area (i.e., histogram bin) represents the particle concentration in this area. The three blue symbols denote, from lower left to upper right, a ground sensor network, a UAV, and a Home Guard patrol.

2.2.2. Terrain dependent motion and birth model

The state of a vehicle hypothesis at time t depends (Eq. (20)) on the state of the hypothesis at the previous time-step $t-1$, and on the terrain at the vehicle position $\mathbf{y}_t$. Likewise, the state of a newly born particle (Eq. (21)) depends on the observation from which it was born, and on the terrain at its position.

While the dependence on previous time can be modeled using linear dynamics, the terrain dependence is highly non-linear. For each position, the terrain can be retrieved from the database (see Section 3.4). The terrain influence on the vehicle position is represented as probability ratios $\pi_{\text{water}} = p_{\text{water}}/p_{\text{road}} = 0$, $\pi_{\text{forest}} = p_{\text{forest}}/p_{\text{road}} = 0.04$, $\pi_{\text{field}} = p_{\text{field}}/p_{\text{road}} = 0.2$, and $\pi_{\text{road}} = 1$ of the vehicle being positioned in different type of terrain.

The sampling from the conditional pdf $f_{\mathbf{X}_t|\mathbf{X}_{t-1}}(\mathbf{x}_t|\mathbf{x}_{t-1})$ is performed in two steps:

1. Each particle in the old posterior cloud $\{\xi^1_{t-1},\ldots,\xi^{n_{t-1}\mathcal{N}}_{t-1}\}$ is propagated using a first order linear dynamical motion model.
2. Each new particle is given a weight $\pi_{\text{terrain type}}$ depending on the terrain type at its position. A new particle cloud is then Monte Carlo sampled from the weighted particles.

Likewise, the sampling from the conditional pdf $f_{\mathbf{X}_t|\mathbf{z}_{t-1}}(\mathbf{x}_t|\mathbf{z}^i_{t-1})$ for each old observation $\mathbf{z}^i_{t-1}$ is performed as

1. $\mathcal{N}$ particles are sampled from observation $\mathbf{z}^i_{t-1}$ using a linear Gaussian model. The cloud is propagated using a first order linear dynamical motion model.
2. Identical to step 2 above.

2.2.3. Sensor position dependent detection rate

The probability of missed detection p_{FN} varies over space and time, due to the type and fields of view of the different sensors.

To achieve a correct PHD estimate it is important to model this variance. For each sensor i in the system, the target detection probability p^i_t and the present field of view A^i_t is known at a given time-step t (see also Section 3.1). The probability of missed detection in a certain position $\mathbf{y}$ can then be derived as

$$p_{FN}(\mathbf{y}) = \prod_{\mathbf{y}\in A^i_t}(1-p^i_t). \tag{26}$$

This varying p_{FN} is used for propagation of the PHD over time as described in Eq. (24).

This is intuitively obvious: if there are no sensors nearby, the prior particle distribution is accepted as posterior distribution as is. However, prior particles that come inside the field of view of a sensor are suppressed if there are no observations to support them. Accurate sensors with high p^i_t suppress particles to a higher degree than sensors with a low p^i_t.

2.3. Sensor allocation

The allocation module in IFD03 implements a simple version of sensor allocation based on *random set simulation*. As in the tracking module (Section 2.2), random sets [39,11] are used to formally describe the algorithm's operation, and the probability hypothesis density is used to render the method computationally feasible.

The purpose of the sensor allocation method implemented in IFD03 is to determine which of several sensor allocation schemes should be used in a given tactical situation. In the demonstrations, sensor allocation is performed when the commander wants to determine which of three possible roads a previously observed heavy tank company will take. Input to the module are a list of such allocation schemes or plans, a road network that describes the geography of the situation of interest, and estimated positions of enemy units. The positions are extracted from the other analysis modules. The road network was constructed by computer processing of a list of coordinates for roads extracted from the FLAMES database. Sensor allocation schemes were constructed by hand.

In order to determine the best of the sensor allocation schemes, we simulate a possible future path of the enemy units and apply all the sensor schemes to it. For each sensor scheme, this gives a list of simulated observations that can be input to a fusion module. For each fusion output, we calculate a fitness value by comparing it to the "true" simulated future. By averaging over possible future paths, we determine a total fitness for each sensor allocation scheme.

Mathematically, the algorithm works as follows. A density vector x_0 is given, which describes the positions of the units of interest at time $t=0$. A set S is defined, consisting of sensor allocation schemes and information about the road network on which the enemy is assumed to move.

Three different random sets are introduced:

1. $\mathbf{X}(t)$ denotes the positions of the enemy units at time t, conditioned on their being located at $\mathbf{x}_0$ at time 0. It can be seen as representing a simulation of ground truth: the instance $x(t)$ of $\mathbf{X}(t)$ occurs with probability $P[\mathbf{X}(t) = x(t)|\mathbf{X}(0) = x_0]$. For simplicity of notation, the conditioning on x_0 is not explicitly shown in the following.
2. For each sensor allocation scheme $s \in S$ and instance $x(t)$ of the future ground truth, a set of possible observations $\mathbf{Z}(x(t), s, t)$ is calculated at time t. $\mathbf{Z}$ is also a random set; note that it depends on ground truth as well as on allocation scheme.
3. Finally, we determine what our view of ground truth would be, given the set of observations $\mathbf{Z}$. This gives rise to the final random set, $\mathbf{Y}(t)$. $\mathbf{Y}(t)$ is our fusion system's approximation of the (simulated) ground truth $\mathbf{X}(t)$ using the observations $\mathbf{Z}$ obtained by deploying sensors according to sensor allocation scheme s_i.

All of the random sets introduced are explicitly time-dependent. Here, an expression like $P[\mathbf{X}(t)]$ denotes the probability of the entire time-evolution of $\mathbf{X}(t)$, not just the probability at a specified time. $P[\cdot]$ can thus be seen as a probability density functional in the space of all explicitly time-dependent random sets.

Determining which sensor allocation scheme to use is now done simply by comparing the assumed ground truth $x(t)$ to the fusion system's simulated view $y(t)$. For each instance $x(t)$ of $\mathbf{X}(t)$, the best s can easily be determined by averaging over the ensembles of observations $\mathbf{Z}$ and simulated filter output $\mathbf{Y}$ entailed by that simulated ground truth. An allocation scheme is good if the simulated filter gives a good approximation of the simulated ground truth. The fit of a specific allocation scheme s for a certain simulated ground truth $x(t)$ can be written as

$$\begin{aligned} H(x(t), s) = \int & P[\mathbf{Z}(t) = z(t)|\mathbf{X}(t) = x(t), s] \\ & \times P[\mathbf{Y}(t) = y(t)|\mathbf{Z}(t) = z(t)] \\ & \times h(x(t), y(t))\, \mathrm{d}z(t)\, \mathrm{d}y(t) \end{aligned} \tag{27}$$

where h is a functional that compares $x(t)$ and $y(t)$ and the integrals are functional integrals over all random sets $y(t)$ and $z(t)$. In IFD03, four different h-functionals are used: two which compute the entropy of y at either a user-specified target-time or averaged over all time, and two which calculate the L_2 distance between x and y, again either at a specific time or averaged over all times. The difference between the entropy-like measure and the distance measure is that the entropy measure rewards allocation schemes that give rise to peaked distributions, but might miss some of the enemy units. A measure that uses a specific time is termed a local measure, while global h-measures average over all times.

The overall best sensor allocation scheme is then determined by averaging also over the random set $\mathbf{X}(t)$, as

$$s_{\text{best}} = \arg\min_{s \in S} \int P[\mathbf{X}(t) = x(t)] H(x(t), s) \mathrm{d}x(t) \tag{28}$$

Implementing Eqs. (27) and (28) would thus entail averaging over three different random sets, which is clearly computationally infeasible. There are several possible ways of approximating these equations.

One way is to use approximations of the probabilities P appearing in them, perhaps employing some kind of Monte Carlo sampling instead of the ensemble averages. In the implementation used in IFD03, we use a number of approximations:

1. As stated above, all motion of adversary units is constrained to a road network. Also, discretised time is used instead of continuous.
2. Instead of full random sets for simulated ground truth, observations, and simulated filter, PHD's are used for these. This means that, for instance, $x(t)$ only gives the expected number of units at different positions in the road network.
3. A very simple model is used for determining $P[\mathbf{X}(t) = x(t)]$ and averaging over all $x(t)$: it is assumed that the adversary's movement can be described by a motion model $\mathbf{M}$. This model is used to determine paths for all adversary units present at time $t = 0$. Instead of averaging over all possible futures, a certain number N_f of such paths are generated and assumed to have equal probabilities of occurring.
4. A similar motion model in the form of a transition matrix $\mathbf{M}$ is used to simulate the filter determining $\mathbf{Y}$, and we average only over a number N_o of possible observations (i.e., realizations of $\mathbf{Z}$).

Pseudocode for the sensor allocation algorithm is shown in Fig. 9.

The sensor allocation module returns the best found sensor allocation scheme $\tilde{s}_{\text{best}}$ as well as a quality measure that simply gives the fraction of the number of simulated ground truths and observations in which $\tilde{s}_{\text{best}}$ dominated all other allocation schemes. In experiments performed using IFD03, the allocation scheme selected was also the one which a human analyst would choose, given the same information as the method.

There are several directions of future work related to the method described here. Computational efficiency can be improved considerably by considering equivalence classes of future paths for the enemy units. The concept of such equivalence classes is described in detail in [40]. Briefly, we consider two paths equivalent if they give rise to the same set of observations, and we only need to average over all equivalence classes when determining the fitness of a sensor scheme. The computational cost could also be reduced by using Monte Carlo sampling instead of the simple averaging over the three random sets used here.

```
% Pseudo code for the four major steps of the allocation algorithm.
% The module simulates N_t ground truths and does
% N_o realizations of the observation process for each
% ground truth, averaging the fitnesses.  This process is
% repeated for each sensor allocation scheme s, and the best s
% is selected.
% Note that x_t, y_t and z_t here are vectors;
% we have discretised space to only include nodes that
% are present on the road network.  M and δ below
% represent the motion model constrained to this network.
% t_end is the end time of simulation, while p_d is the
% assumed detection probability of a sensor.

% Simulating ground truth:
x_1 = x_0
for t = 1 : t_end − 1
    x_{t+1} = x_t + δ where δ is randomly selected
end for

% Generate fictitious observations:
for t = 1 : t_end
    if s has sensor with view of x_t at time t
       generate observation z_t = x_t with probability p_d
    end if
end for

% Simulate filter:
y_1 = x_1
for t = 1 : t_end − 1
    y_{t+1} = M y_t + z_{t+1}
    where M is a transfer matrix corresponding to the road network
end for

% Compare simulated ground truth and filter:
h_1(s) = ||y_{t_end} − x_{t_end}||_2
h_2(s) = Σ_t ||y_t − x_t||_2
h_3(s) = H(y_{t_end})
h_4(s) = Σ_t H(y_t)
where H(x) is the entropy of the vector x
```

Fig. 9. Pseudocode for sensor allocation.

Furthermore, more work needs to be done on automatically generating sensor allocation schemes. This could be done by, e.g., generating all possible UAV paths that cover the road network or using genetic algorithms to modify paths and evaluate them using the method presented in this section. Another possibility would be to use a swarming or collective intelligence algorithm for this, similar to [41].

The method could also be used interactively by a user, who suggests partial or complete sensor schemes to the system. Here work needs to be done both on how to generate a complete plan from a partial one and on how to best interact with the user.

3. System architecture

In this section, the overall design of the IFD03 system is described. The demonstrator utilizes all of the methods described in Section 2, implemented as "cognitive models", i.e., behavioral submodels of simulated actor models. It also makes use of an advanced terrain database that has been integrated into the simulation framework FLAMES and provides a standard procedure for scenario definition, which can be used to flexibly combine the various object models to form specific scenarios.

An overview of how the different components fit together is given in Fig. 10.

3.1. Scenario simulator

All data originate in the scenario simulator. The primary objects of the simulation fall into three categories: actors, terrain model and fusion node.

- *Actors*. The actors in a scenario simulation consist of "red" (adversary) and "blue" (own) units. Each unit is equipped with a platform model and a radio model for communication. Blue units are also equipped with various sensors for target detection and classification. To

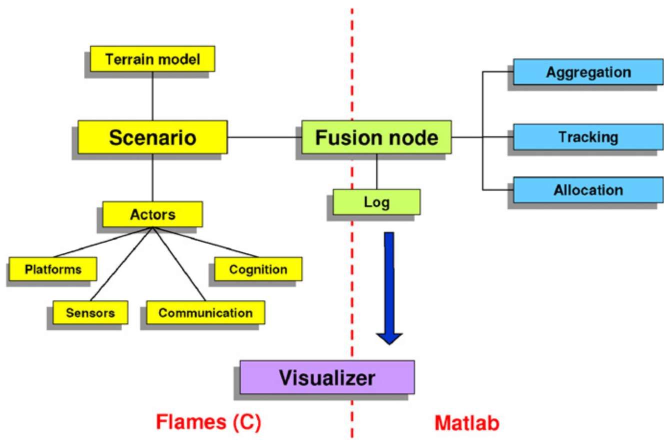

Fig. 10. Connections between different parts of IFD03. Lines between modules mean that the modules exchange data. Note that the Visualizer is a separate program, while all the other modules are linked into the `Fire` program.

enable target detection and classification, visual and acoustic/seismic signatures are attached to all red platforms. When a detection is made, a report is sent to the fusion node with target information.

- *Terrain*. As detailed in Section 3.4, the terrain model represents the features of the environment, such as terrain elevation or vegetation foliage, by polyhedra. The platforms and sensors in a scenario use this information for mobility and visibility calculations. The same information is available to the fusion node, which currently uses it only in the tracking algorithm.
- *Fusion node*. The fusion node has access to a priori information in the form of a terrain model and a doctrine and materiel database, generically describing the adversary's military organization. Also, it has the capability to perform dynamic remote control of a set of sensors which can observe portions of this force. Most importantly, the fusion node provides services for report clustering, aggregation, classification, and tracking of force units, and for allocation and control of information collection resources, see Section 2. The other blue units continuously feed the fusion node with target reports and, upon request by the fusion node, sensor status reports. By sending sensor status reports a unit updates the fusion node about the current coverage of its sensors. This information is used by the tracking algorithm, see Section 2.2.3.

  In a formal sense, the fusion node is also an actor in the scenario. It is implemented as a FLAMES cognitive model attached to a blue unit, modeling those aspects of a C4ISR system which are needed in the application.

The parts of IFD03 which handle scenario simulation are written in C and directly linked into the FLAMES suite of programs. All fusion modules are implemented in MATLAB code, which are first auto-translated into C using the MATLAB Compiler, then wrapped (by hand) as FLAMES cognitive models and finally compiled and linked into the FLAMES program `Fire` to produce an executable.

3.2. Modeling doctrine, organization, and equipment

IFD03 requires models describing the behavior and motion of adversary ground forces according to their doctrine, i.e., the set of tactical rules that is expected to guide the behavior of the opponent's army. This includes telecommunication and transportation along a road network of mechanized forces in hostile territory.

The adversary battalion model consists of approximately 60 vehicles: battle tanks, two types of tracked armed personnel carriers, wheeled antiaircraft missile launch vehicles, and mortar vehicles. To create models of these target objects as well as of a number of additional vehicle types occurring in related force structures, a table of normalized detection, classification, and identification probabilities are needed for each object type and each type of sensor. In these tables, objects are assumed to be viewed at a fixed distance and against a clutter-free image background, noise-free seismic or acoustic environment, etc. Attenuating properties of the environment will reduce these probabilities as they occur in observing situations. Five different battalion structures were included in the force aggregation template database. The descriptions include unit hierarchy down to vehicles of specified types. From these resource descriptions the application "march under low threat" was developed, which includes the sequence of and distance between vehicles and units, from vehicles via platoons to companies.

The information used in modelling the radio communication needed to stimulate COMINT interceptors describes the commanding hierarchy and simple communication rules.

3.2.1. Sensor modeling principles

How a sensor can be modelled depends strongly on its type. In general what is needed is some kind of detection or recognition time for each sensor, e.g., for an image sensor, a shortest time during which an object must be continuously visible to be detected, classified, or identified, each step in this sequence requiring additional time. These times depend on sensor type, obstacles in the line of sight, and target object type, in combination with target attitude in relation to the sensor.

The resolution of an image-generating sensor is vital for the sensor's ability to detect, classify, and identify a target. It depends on optics, zoom factor etc. Additionally, the contrast between light and dark parts of the image has to be strong enough [42]. The object's aspect angles in relation to the observing sensor are also of relevance. Finally, the surrounding environment generates clutter which reduces the sensor's ability to distinguish objects. Factors which are significant in determining the sensor's detection and recognition capabilities should be weighed against each other. These may be grouped into three general categories:

1. A sensor detects energy (light, vibrations, radio signals, etc.).
2. The energy has been emitted from somewhere, as well as propagated and attenuated, reflected etc., on its way to the sensor.
3. For detection to take place, the detected signal-to-noise ratio (SNR) from a target must be high enough for a signature-extracting mechanism to find the features it is trained to discern.

The fusion methods tested in IFD03 expect input such as observation times, target positions and velocities with their uncertainty estimates, as well as target types with uncertainty within a given target classification hierarchy. Recognition type hierarchies relevant for the different sensors' energy classes were constructed. According to these, an image sensor can, e.g., discern (with decreasing discernibility) <T80>|<tank>|<tracked vehicle with 6 track rollers>| <vehicle>, and a seismic sensor can discern <heavy tracked vehicle>|<vehicle>. This allows comparisons based on evidential reasoning to be made between type information from different sensor categories.

Sensors used to detect ground targets will likely show greater rates of false detection the more difficult prevailing surveillance conditions are. In the scenario used, terrain is diversified with many forested areas of different sizes, open croplands, roads, lakes, and littoral regions.

3.2.2. Sensor models

Image sensors. Image sensor detection probability is based on the empirical "Johnson criterion" [43]. It gives a relation between the number of resolved pairs of light/dark bars in a bar pattern of the same size as the target minor extension projected towards the observer, and the probability of detection, classification, or identification. The number of resolved bars is related to the contrast between light and dark regions in the image, also interpretable as SNR. For a sensor in the visible region, this is the contrast of reflected light within and at the bounds of a potential target. This contrast is dependent on target surface reflectance variations, and the strength and direction of the ambient light. For an IR sensor in the thermal region, contrast depends instead on the target and background temperature variations.

The attenuation of light is modelled for an image sensor observing from a UAV. The attenuation factor is dependent on terrain cover type (forest/open land), and, in the forest case, on the angle between the line-of-sight (LOS) and the vertical direction.

Ground sensor networks. A simple ground sensor network model was implemented. It is assumed to possess an integrated tracker, so that terrain effects as well as the influence of the individual positions of the network nodes could be disregarded. This entails a statistically homogeneous detection capability inside the range of the network. Such a sensor is able to contribute high quality position and speed measurements, but only poor classifications.

Human observers. The model of human observers, Home Guard patrols equipped with advanced measuring binoculars, is less detailed than the image sensor discussed above. This is mainly due to difficulties of modelling the complex fusion performed by the human brain. A basic relationship of detection quality proportional to distance was assumed and model parameters were then adjusted so as to produce reasonably realistic output.

COMINT interceptors. Radio messages can be intercepted by blue COMINT interceptors deployed in the terrain. The interceptors give rather coarse information about bearings to emitters. Bearing crossings are computed to get an indication of the position of an emitter. Information about position and communication pattern is transmitted to the fusion node, which tries to find out who is communicating with whom, see Section 2.1.2.

3.2.3. Sensor carriers (platforms)

Sensors are carried by either an unmanned aerial vehicle (UAV), a COMINT interceptor station, or a soldier. A video/IR camera can be attached to the UAV, which can also carry and drop a ground target multi-sensor system.

3.3. Scenario display

During demonstrations, three adjacent projection screens show, respectively:

- reports and ground truth data displayed on a synthetic map background,
- results from the different information fusion methods displayed on map backgrounds, and
- dynamic plots of various statistics and other information about the current state of the fusion processes.

These views are intended to support a tactical intelligence staff in building a situation picture.

At the beginning of the scenario only a few reports have arrived. These are indicated on the first screen (Fig. 11) and then appear as clustered objects on the second screen (cf. Fig. 2). This is the first chain of fusion events shown during the demonstration. At the same time the process can be followed on the third screen where plots of the number of received reports and the estimated number of objects are displayed (Fig. 12).

As the scenario progresses, more surveillance resources are allocated and therefore many more reports are delivered. On the second screen, views showing clustered vehicles and clustered platoons are displayed. Here, vehicles and platoons are automatically classified into more or less specific categories, when possible into specific types. The categories or types are displayed using standardized army symbols (Fig. 2). The operator can switch instantly

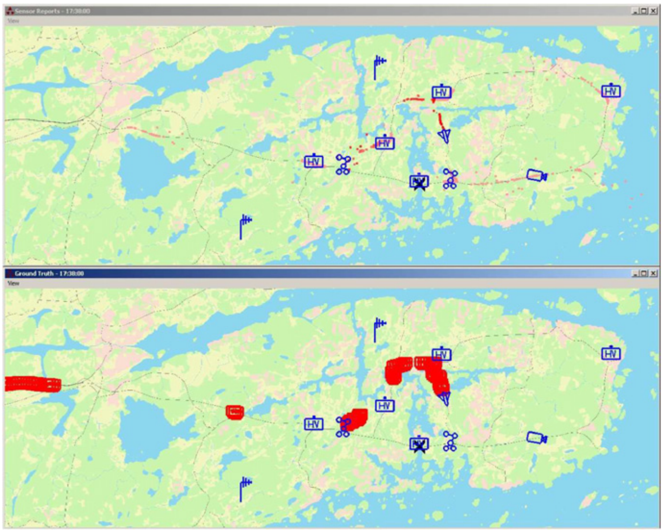

Fig. 11. Snapshot of the sensor report (top) and ground truth (bottom) views in IFD03.

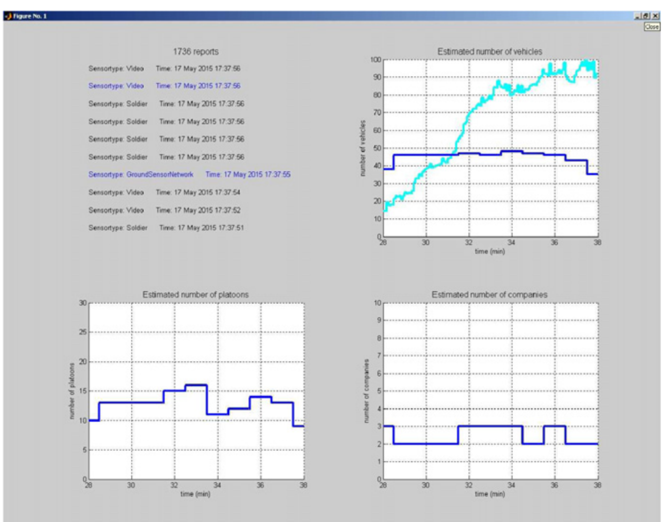

Fig. 12. Snapshot of the standard status view in IFD03. The sub-plots show incoming reports and the estimated number of vehicles, platoons and companies over time.

between different aggregation levels such as reports, vehicles, platoons and companies, showing how the different information fusion methods work at different levels. The display can be paused or moved back or forth in scenario time, to show how correspondences vary in different situations and between different levels of the scenario. By zooming in on any desired display area, detailed situations can be visualized and discussed. Additional information can be obtained by right-clicking on the symbols in a view. An information box then appears with details specific for each type of symbol.

To indicate how the fusion methods are performing, various results can be compared with ground truth while they are displayed. Having access to all scenario information, the ground truth view shows the location of all vehicles and all sensors in the displayed area over time.

3.3.1. Visualizer

The IFD03 Visualizer is a substantially modified version of the original FLAMES visualizer `Flash`. The simulation results can be visualized in multiple parallel visualizers, making it possible to use several computers and screens simultaneously. New views can easily be created and customized.

From a programming perspective the IFD03 Visualizer consists of four entities, three of which are executable applications. The database, handled by a MySQL database manager [44], stores simulation result data to be visualized. The postprocessor application creates tables in the database. It also converts and transfers simulation result data into the database. The playback control application synchronizes the playback of the scenario across the different connected visualizers. Finally, the modified `Flash` application performs the actual visualization of the data.

3.4. Environment model

The terrain model for IFD03 is created by using a third party terrain database generation tool, TerraVista Pro Builder [26], that can import data from different sources and export a single correlated, i.e., geometrically and topologically consistent, terrain description. TerraVista also has the ability to write the correlated data in a variety of formats. The terrain model is structured as a TIN DEM (Triangulated Irregular Network Digital Elevation Model) representing the terrain skin, and additional vector data describing terrain features, such as roads, rivers, lakes and houses (Figs. 13 and 14).

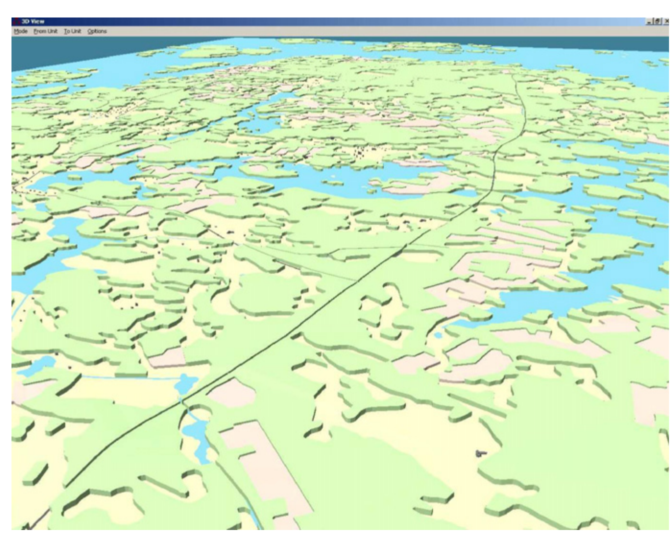

Fig. 13. Visualizer perspective eastward, covering a part of the scenario area of interest. The terrain is color-coded according to (forest: green, open land: yellow, water: blue, road: grey, built-up areas: pink). Note the different heights of forest canopies and built-up areas.

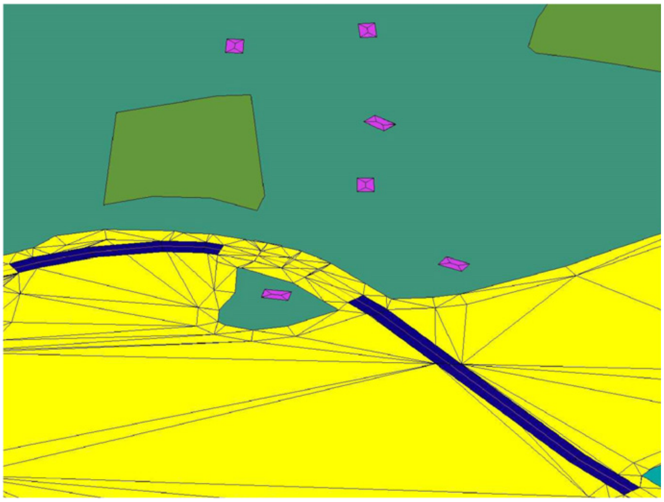

Fig. 14. Terrain elevation and feature data after adding roofline data for six buildings (purple) and areal data for two bridges (dark blue).

3.4.1. Terrain data

The source data for this terrain model consisted of conventional off-the-shelf geographic data from the Swedish Land Survey. Source data used in the project come from their GSD (*Geographical Data of Sweden*) product. For the scenario used in IFD03, data describe a $45 \times 20\ \mathrm{km}^2$ area including the peninsula of Rådmansö, north-east of Stockholm. The terrain features were grouped into seven classes defined by FLAMES: bridge, building, canopy, land region, lake, river, and road. The $50 \times 50\ \mathrm{m}^2$ ground elevation database from which our triangulated 3D model was built is probably detailed enough for our current needs, although for realistic modelling of the strongly varying tree canopy transparency, one would probably need, say, $25 \times 25\ \mathrm{m}^2$ raster information on tree population density and type, as well as typical tree height and mixing ratio of coniferous and deciduous trees.

4. Fusion performance evaluation

The performance of the three fusion methods described in Section 2 was evaluated as follows. The force aggregation method was qualitatively evaluated by inspecting results from the demonstration scenario. Vehicle and platoon aggregation results were studied, originating from a set of situations with different characteristics. As detailed in Section 2.1, aggregation results derive from conflict-based Dempster–Shafer clustering and classification of sensor reports giving vehicle position, direction, and type estimates, followed by clustering of vehicle estimates and classification of the resulting "platoon" clusters using organizational template matching. The performance of the terrain tracker was studied during a small part of the scenario, in which a mechanized company left the road to travel across a stretch of half-open terrain. Evaluations of these methods used data saved in the MySQL database during a previously performed simulation of the scenario described in Section 1.4. Finally, the sensor allocation method was evaluated using an extensive simulation experiment originally developed to verify a refined version of the method, see [40].

4.1. Evaluating the force aggregation method

4.1.1. Case studies

We have selected four aggregation situations for analysis and presentation, two of them close in space and time. In all cases, we study the result of both vehicle and platoon aggregation. Company aggregation was not analyzed. During our result analysis we studied several other situations in detail. Our general conclusions below are based one these situations as well.

In Fig. 15 we show first a vehicles estimate obtained by clustering of sensor reports, where 10 observed ground truth vehicles have generated 11 correctly but sometimes imprecisely classified vehicle estimates. Eight of these were also precisely classified. One additional precisely classified vehicle estimate belongs to a previously passing unit. The platoons estimate grouped 9 of the 10 vehicles correctly, 3 by 3, into correctly classified platoons. However, the remaining two vehicle estimates, one of them based on old observations, were combined into a type-specific but actually incorrect platoon. If the old vehicle estimate could have been eliminated, this type of problem would not occur, cf the discussion below.

In Fig. 16 we show another situation in an area where no other vehicles have previously passed by. Fifteen ground truth vehicles generated 13 correctly classified vehicle estimates, 7 of these precise. One precisely classified vehicle estimate was incorrect, due to an erroneous specific sensor report being combined with other non-specific reports. The platoons estimate grouped the estimated vehicles into 5 correctly classified platoons, one too many.

In Fig. 17 we show a situation where a set of vehicles (a company) are leaving the road and moving into the terrain. Here, 15 ground truth vehicles generated 15 correctly classified vehicle estimates, 9 of these precise. The platoons estimate grouped the estimated vehicles into 2 correctly classified platoons. This is 3 platoons less than ground truth. In this case, the speed is being reduced because of the move into terrain, creating a queue with shorter intra-vehicle distances than is typical for road travel. The maximum distance between vehicles assumed in the aggregation algorithms is constant and adapted to normal on-road movement, making it possible for the clustering algorithm to pack more vehicles into each platoon in this atypical situation. Since the clustering algorithm uses only binary relations between objects when deciding which objects may be clustered, it is unable to exploit a priori knowledge about the maximum vehicles allowed in a platoon of a certain type. The only information it can use is the maximum and minimum number of platoons allowed, as given by the current number of vehicles

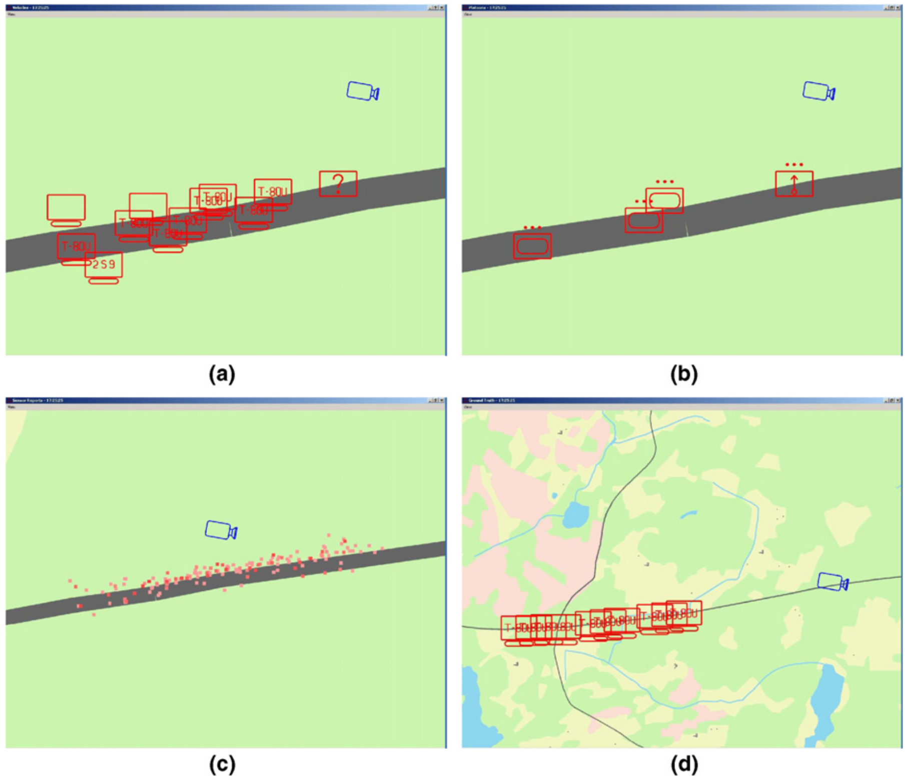

Fig. 15. Aggregation results at the roadside camera. Time is 17:25:25 (hh:mm:ss). Vehicle ground truth and sensor reports views are shown for comparison. (a) Vehicles estimate. (b) Platoons estimate. (c) Sensor reports. Color becomes less saturated as reports are ageing. (d) Vehicle ground truth.

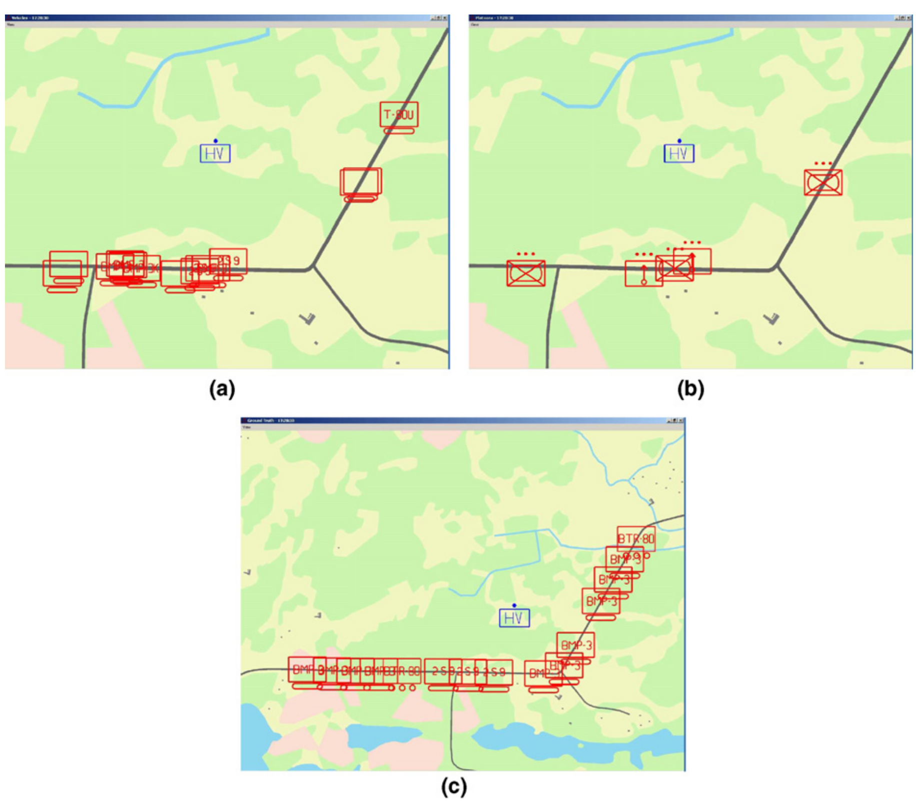

Fig. 16. Aggregation results from Home Guard observations of vehicles on the northern road. Time is 17:28:30. The vehicle ground truth view is shown for comparison. (a) Vehicles estimate. (b) Platoons estimate. (c) Vehicle ground truth.

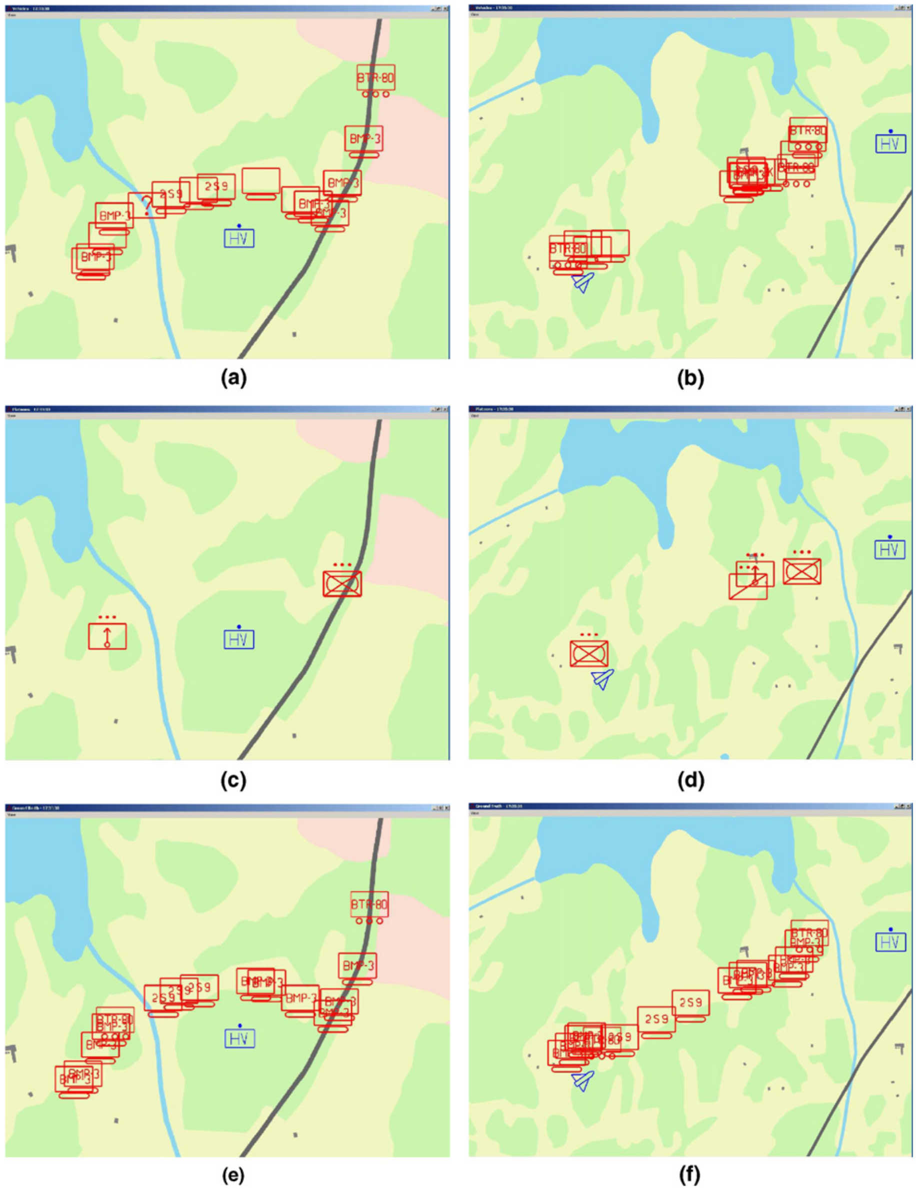

Fig. 17. Aggregation results from Home Guard observations of vehicles (a mechanized company) coming south from the northern road and moving into the terrain. Left column data are from 17:33:30, right column data from 17:35:30. A UAV is flying by at the latter instant, contributing additional observations. (a) Vehicles estimate at 17:33:30. (b) Vehicles estimate at 17:35:30. (c) Platoons estimate at 17:33:30. (d) Platoons estimate at 17:35:30. (e) Vehicle ground truth at 17:33:30. (f) Vehicle ground truth at 17:35:30.

divided, respectively, by the minimum and maximum number of vehicles allowed for any platoon template.

4.1.2. Discussion

The most easily interpreted results were obtained from sensor observations of the moving front of the advancing battalion. Given that available sensor data were sufficiently specific and complete, in this case the method usually performed well, see Fig. 16.

In some situations all available sensor data (usually from UAV observations) were highly unspecific, in others a large number of sensor reports covered an interesting area, including some which were several minutes old and therefore indicated vehicles which had long since left the area.

In the first case the vehicle clustering and classification methods generate a set of unspecific vehicle observations. The aggregation method deduces the organization type by combining the observed vehicles types and the number of observed vehicles. When the observed number of vehicles is smaller than the true one and no type information is available, the visualizer may deliver a specific unit type determination based only on the estimated (but too low) number of vehicles. Only the most likely alternative is currently displayed although basic probability numbers weighted by platoon-to-template fitness are calculated for all alternatives. A more informative approach might be to inspect the structure of this set and decide whether it is appropriate to display a specific result, or whether to opt for an unspecific one.

In the second case, note that a vehicle moving at the typical scenario speed 50 km/h will have traversed more than 8 kilometers in 10 min, a large fraction of the entire scenario transportation distance. Thus, in this scenario, vehicles observed several minutes ago do not actually belong to the current local scene. In a future decision support system space and time windows used for aggregation should be adjustable by the user, or if possible, be automatically adapted to the situation. A simple but useful improvement would be to color-code the age of estimated objects in all views, a feature now used only in the sensor report views, see Fig. 15.

The way it is currently used, the aggregation method is based on only discriminating information. On the report-to-vehicle level it rejects configurations it considers impossible on grounds of speed limitations, directions, and type inconsistencies. On the vehicle-to-platoon level it rejects configurations based on track-to-track distances and type inconsistencies, see Section 2.1. Using also attractive information when available could improve aggregation performance in certain situations.

In some cases, the computed distribution of platoon types was quite far from the correct one. To correct this, several approaches could be tried:

1. Trimming the existing criteria.
2. Introducing fuzzy templates which do not penalize partially observed units.
3. Introducing additional kinds of criteria, such as top-down analysis of consequences of the assumed organizational structure. This could lead to adjustments of lower level (e.g., vehicle type) estimates.
4. Multi-hypothesis reasoning, deriving a set of alternative, incrementally different clustering results, to be used as alternative options in the platoon-to-template matching process.

4.2. Evaluating the terrain tracking method

The discussion below of the performance of the particle filtering method is based mainly on observations during the terrain passage shown in Fig. 18, to be compared also with the right column of Fig. 17.

Judging the utility and performance of a PHD particle filter in this application is subjective, its outcome depending strongly on how results are to be used. Here a method was needed for short-term qualitative prediction of vehicle movements in terrain and on roads. Quantitative use of the method, e.g. to precisely predict target locations, was not prioritized. In Fig. 18, we illustrate how the PHD particle filter satisfies this rather modest requirement. The scenario situation is such that after the fly-by of the UAV at 17:35:35, no more direct observations were possible until some part of the unit reached the area of sensitivity of the ground sensor network in the lower left of the depicted area. Two minutes later, on average the filter predicts a somewhat more easterly path for the unit than was actually used, Fig. 18(b). After one more minute, the prediction has become quite unreliable, Fig. 18(c). At that moment the ground sensor network has however already picked up signals from vehicles at the front of the unit, gradually allowing the filter's prediction to recover.

Thus, continued filtering without any new observations gave a reasonable position estimate for no more than two minutes in this situation. The filter parameters could probably be adjusted to better meet a user's need of seeing the distribution of possible locations for the unit rather than focusing on trajectories considered to be the most likely.

4.3. Evaluating the sensor allocation method

In order to evaluate the sensor resource allocation method described in section 2.3, experiments were performed where the fitness of a large number of sensor allocation schemes was determined. Since it is difficult to define a true fitness for a sensor scheme, the relative ranks of the schemes was then subjectively compared to ensure that they were consistent with a human operator's ranking. Recall that in IFD03, a certain number of predetermined sensor allocation schemes were input to the sensor allocation module, which determined the fitness of each of these. The schemes used in IFD03 were designed so that one of them should be clearly better than the others, one should be clearly worse than the others and the rest have about equal fitness. An additional 10^4 sensor allocation schemes[2] were generated by mutating the IFD03 plans in the following way:

- UAV paths were cut into small sub-routes that could be connected independently of each other,
- places and times to drop ground sensor networks were determined randomly.

The UAV routes were cut and the sub-routes merged since we did not want to implement a complete route-planning algorithm for UAVs.

The fitnesses of each of these sensor schemes s were determined using the four fitnesses h_1, h_2, h_3, and h_4 from Fig. 9. In order to validate the evaluation method, the results of $h_i(s)$ were inspected both for specific instances $x(t)$ of the simulated ground truth $X(t)$ and for the averaged values. Since most of the sensor schemes used in IFD03 were hand-designed to be good, we expected all the randomly generated schemes to be worse than those. This expectation was fulfilled in the experiment: unsurpris-

[2] We ran the randomization process 10 times generating 10^3 different sensor schemes each time.

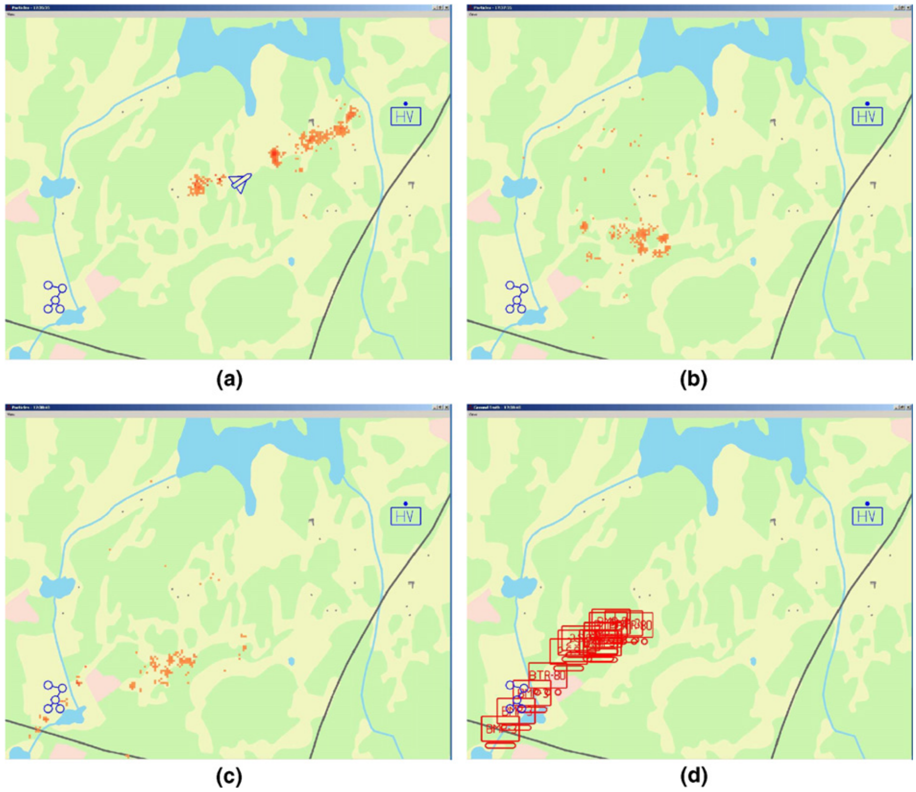

Fig. 18. Results from particle filter tracker during the terrain passage, cf. Fig. 17. Vehicle ground truth at 17:38:40 is shown for comparison. (a) Particle histogram at 17:35:35. (b) Particle histogram at 17:37:35. (c) Particle histogram at 17:38:40. (d) Vehicle ground truth at 17:38:40.

ingly, most of the randomly generated schemes were very bad. After determining the ranking of the schemes, the best ones and some randomly selected bad ones were checked manually to make sure that there were no obvious misses in the ranking induced by the fitness calculations. For a more thorough validation of a more advanced sensor scheme evaluation method implemented using equivalence classes of future paths, refer to [40].

5. Comparison with other approaches

The research that has been devoted to information fusion issues since Kent's paper [12] was written has advanced the state-of-the-art in methodological areas likely to become important in future command and control applications. At the same time, however, requirements and ambitions have expanded substantially, indicating that tactical application of information fusion is not a well-delimited problem that will be solved once and for all, but a growing research subfield of applied artificial intelligence with the potential to successively provide new user benefits far into the future. At the current state of knowledge, researchers need to explore and compare several alternative methodological approaches while watching out for particular emerging techniques suitable for early transfer to fieldable systems. Below, a few such approaches are briefly addressed and compared with those developed for IFD03.

An early paper discussing the processing of tactical information and the associated situation assessment of the tactical battlefield is [45]. The architecture proposed and elaborated in this paper by Gonsalves et al. combines fuzzy logic and Bayesian belief networks (BN) for constructing and maintaining a hierarchical, probabilistic model linking multiple entities at various levels, in the context of overall mission goals and rules of engagement. Evidence gathered incrementally in real-time undergoes fuzzy logic filtering and is then applied to the appropriate nodes of the BN, resulting eventually in revised probability estimates concerning tactical situational hypotheses. A more recent contribution to this class of methodologies is a paper by Sutton et al. [46], where they propose a blackboard BN architecture, enabling analyst users to manipulate and combine BN fragments into hypothetical models of interpretation which may then be evaluated using both observational and a priori inputs. Knowledge about the world necessary for creating meaningful models is provided by both analysts and the "corporate memory" stored on the blackboard. The architecture extends blackboard techniques with a principled method (BN) for representing uncertainty, and it extends Bayesian network techniques by a facility for incremental model-building.

One of the few papers which provide a commander's perspective on information fusion, emphasizing the need for a multi-role capability, is written by Looney [47]. In this paper Looney focuses primarily on what he calls an "alternative" methodology for level 1 fusion, although he also makes interesting remarks on requirements for higher-level fusion methods in common operational picture (COP) applications. The paper concludes with the statement: "The major outstanding problems are in SA (situation assessment) and TA (threat assessment) and the monitoring and adjusting of the fusion system to reduce errors. However, there remain many problems in the detection, tracking and identification of targets of all types so that new tracking algorithms are needed". In a later paper, Looney and Liang [48] return to the question of providing tools to support building up and maintaining situation awareness for tactical commanders. They propose using the centralized *k*-means clustering algorithm in a case-based reasoning (CBR) context, to obtain robust unit type, size and purpose determination on several levels of abstraction. These are then fed into a fuzzy belief network that performs inferencing for threat assessment via heuristic belief propagation.

IFD03 is based on fixed-structure but generic mathematical models, which may be adapted to different situations mainly by exchanging model data provided a priori. Models based on such approaches to BN architectures which do not possess any a priori structure should be highly flexible and may in fact be particularly useful in intelligence applications in asymmetric warfare or operations other than war, where the opponent's order of battle may not be known or even well-defined. These applications, currently widely prioritized, were not addressed by the IFD03 project. However, the flexibility of BN comes at a cost: it is a model-building language having certain intrinsic limitations, rather than a set of ready-made models. Indeed, direct comparison of a language like BN with a system like IFD03 does not make sense.

In comparison with the methods used in [48], the report association of IFD03 can be considered to be more versatile, since it takes into account not only target positions, but all relevant target features and target-to-target relational features. Force aggregation is then performed by Dempster–Shafer template matching rather than CBR. Furthermore, IFD03 uses efficient algorithms, capable of providing accurate answers to difficult computational problems of realistic size sufficiently quickly for practical application. Finally, the problems it solves are ubiquitous in COP applications, particularly in conventional warfare situations. However, no threat assessment is performed by IFD03. In summary, the fusion methods of IFD03 extend the state-of-the-art in providing solutions to a basic set of tactically important problems.

6. Conclusions and future work

We described and discussed above the architecture, methodology and user interface of a software system by which it is possible to credibly demonstrate the use of information fusion for improving commanders' tactical situation awareness in an NBD environment. We showed how the system can be applied to a concrete scenario and discussed how it produces and presents level 1, 2 and 4 fusion results relating to this scenario.

Within the IFD03 demonstrator project, a new method of force aggregation based on Dempster–Shafer clustering and template matching was developed, implemented and tested. Furthermore, a new method for multi-sensor, multi-target tracking of an unknown number of vehicles moving in terrain was proposed, developed and realized. Finally, a new, admittedly preliminary methodology for sensor resource management was proposed, implemented and tested. Although the two first-mentioned methods have been previously published, their implementation and application in a complex and realistic scenario has not been discussed before.

The demonstrator contains a small number of fusion methods based on fixed-structure, generic models, which solve an important set of problems ubiquitous in conventional warfare tactical situations. The models developed for force aggregation and terrain tracking in IFD03 are based on well-founded mathematical theories and use efficient algorithms. Based on experience from our demonstrator as well as on theoretical considerations we claim that these methods represent a substantial advance in comparison to previously reported techniques for situation assessment problems [45,18,48]. In fact, we believe them to be mature enough to be considered for inclusion in a tactical command and control system.

In the project a COTS-based simulation platform tightly coupled to the "fusion engine" was created, enabling the representation of closed-loop management of sensor resources. It has a simulation engine capable of efficiently managing a battalion-sized ground force moving on roads or in terrain, as well as several sensor platforms concurrently.

Based on such demonstrator systems, processes and methodologies can be straightforwardly developed for presenting, visualizing, and analyzing properties of proposed new information fusion components and subsystems. In addition, such systems may be used for performing studies and experiments with sensor models, terrain and other environment models, doctrine models, scenario assumptions, etc. Finally, it can be used to support the development of methodology and models for information fusion per se, i.e., the specification, development, and testing of new fusion concepts and methods.

Further development of force aggregation along the lines discussed in Section 4.1.2 should improve classification accuracy and avoid the propagation of classification errors to higher levels in the organizational hierarchy. The clustering and aggregation techniques may eventually be combined with particle filtering to permit concurrent tracking of both solid objects (e.g., vehicles) and group objects (e.g., ground force units), logically connected via uncertain information about doctrinal rules and communication capability. Work is ongoing [49,50] to develop new methods in the area of sensor resource management that should improve the ability to maintain an accurate operational picture. Finally, studies are being made which should lead to some capability to automatically recognize and predict tactical plans and intentions [51,52].

References

[1] S. Ahlberg, P. Hörling, K. Jöred, C. Mårtenson, G. Neider, J. Schubert, H. Sidenbladh, P. Svenson, P. Svensson, K. Undén, J. Walter, The IFD03 information fusion demonstrator, in: P. Svensson, J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion, Jun 2004, vol. II, International Society of Information Fusion, Mountain View, CA, USA, 2004, pp. 936–943.

[2] P. Hörling, V. Mojtahed, P. Svensson, B. Spearing, Adapting a commercial simulation framework to the needs of information fusion research, in: Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, USA, International Society of Information Fusion, Mountain View, CA, USA, 2002, pp. 220–227.

[3] P. Svensson, P. Hörling, Building an information fusion demonstrator, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 1316–1323.

[4] J. Schubert, C. Mårtenson, H. Sidenbladh, P. Svenson, J. Walter, Methods and system design of the IFD03 information fusion demonstrator, in: CD Proceedings of the Ninth International Command and Control Research and Technology Symposium, Copenhagen, Denmark, paper 061, track 7.2, US Dept. of Defense CCRP, Washington, DC, USA, 2004, pp. 1–29.

[5] A. Steinberg, C. Bowman, Revisions to the JDL data fusion model, in: D.L. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press, Boca Raton, FL, USA, 2001, pp. 2:1–19 (Chapter 2).

[6] J. Schubert, Evidential force aggregation, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 1223–1229.

[7] J. Schubert, Clustering belief functions based on attracting and conflicting metalevel evidence, in: B. Bouchon-Meunier, L. Foulloy, R.R. Yager (Eds.), Intelligent Systems for Information Processing: From Representation to Applications, Elsevier Science, Amsterdam, Netherlands, 2003, pp. 349–360.

[8] J. Schubert, Clustering belief functions based on attracting and conflicting metalevel evidence using Potts spin mean field theory, Information Fusion 5 (4) (2004) 309–319.

[9] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings F (Radar and Signal Processing) 140 (2) (1993) 107–113.

[10] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter: Particle filters for tracking applications, Artech House Publishers, Norwood, MA, USA, 2004.

[11] R. Mahler, An Introduction to Multisource–Multitarget Statistics and its Applications, Technical Monograph, Lockheed Martin Corp., Eagan, MN, USA, 2000.

[12] J. Kent, Deploying tactical fusion systems: the challenges, in: Proceedings of the Third International Conference on Information Fusion, Paris, France, International Society of Information Fusion, Mountain View, CA, USA, 2000, pp. ThB1:11–17.

[13] J. Cantwell, J. Schubert, J. Walter, Conflict-based force aggregation, in: CD Proceedings of the Sixth International Command and Control Research and Technology Symposium, Annapolis, MD, USA, track 7, Washington, DC, US Dept. of Defense CCRP, USA, 2001. pp. 1–15.

[14] N. Xiong, P. Svensson, Multi-sensor management for information fusion: issues and approaches, Information Fusion 3 (2) (2002) 163–186.

[15] D.L. Hall, A. Steinberg, Dirty secrets of data fusion, in: D.L. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press, Boca Raton, FL, USA,, 2001, pp. 21:1–12 (Chapter 21).

[16] R. Mahler, Random-set theory for target tracking and identification, in: D.L. Hall, J. Llinas (Eds.), Handbook of Multisensor Data Fusion, CRC Press, Boca Raton, FL, USA, 2001, pp. 14:1–30 (Chapter 14).

[17] R. Mahler, T. Zajic, Multitarget filtering using a multitarget first-order moment statistic, in: SPIESignal Processing, Sensor Fusion and Target Recognition X, Vol. 4380, SPIE, Bellingham, WA, USA, 2001, pp. 184–195.

[18] F.P. Lorenz, J. Biermann, Knowledge-based fusion of formets: discussion of an example, in: Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, USA, International Society of Information Fusion, Mountain View, CA, USA, 2002, pp. 374–379.

[19] D.A. Lambert, Grand challenges of information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 213–220.

[20] I. Jacobson, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley Publishing Co., Reading, MA, USA, 1993.

[21] J.O. Coplien, Reevaluating the architectural metaphor: toward piecemeal growth, IEEE Software 16 (5) (1999) 40–44.

[22] J. Fredriksson, P. Svensson, T. Risch, Mediator-based evolutionary design and development of image meta-analysis environments, Journal of Intelligent Information Systems 17 (2–3) (2001) 301–322.

[23] D.W. Walker, O.F. Rana, M. Li, M.S. Shields, Y. Huang, The software architecture of a distributed problem-solving environment, Concurrency: Practice and Experience 12 (15) (2000) 1455–1480.

[24] MATLAB, Online, October 2005. Available from: <http://www.mathworks.com>.

[25] FLAMES, Online, October 2005. Available from: <http://www.ternion.com>.

[26] TerraVista, Online, October 2005. Available from: <http://www.terrex.com>.

[27] J. Schubert, On nonspecific evidence, International Journal of Intelligent Systems 8 (6) (1993) 711–725.

[28] M. Bengtsson, J. Schubert, Dempster–Shafer clustering using Potts spin mean field theory, Soft Computing 5 (3) (2001) 215–228.

[29] J. Biermann, HADES–a knowledge-based system for message interpretation and situation determination, in: A. Pasqual del Pobil, J. Mira, M. Ali (Eds.), Tasks and Methods in Applied Artificial Intelligence, Proceedings of the Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, LNCS, vol. 1416, Springer-Verlag, Berlin, Germany, 1998, pp. 707–716.

[30] J.K. Johnson, R.D. Chaney, Recursive composition inference for force aggregation, in: Proceedings of the Second International Conference on Information Fusion, Sunnyvale, CA, USA, International Society of Information Fusion, Mountain View, CA, USA, 1999, pp. 1187–1195.

[31] A.P. Dempster, A generalization of Bayesian inference, Journal of the Royal Statistical Society B 30 (2) (1968) 205–247.

[32] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ, USA, 1976.

[33] R.B. Potts, Some generalized order–disorder transformations, Proceedings of the Cambridge Philosophical Society 48 (1952) 106–109.

[34] P.M. Chaikin, T.C. Lubensky, Principles of condensed matter physics, Cambridge University Press, Cambridge, UK, 1995.

[35] C. Peterson, B. Söderberg, A new method for mapping optimization problems onto neural networks, International Journal of Neural Systems 1 (1) (1989) 3–22.

[36] S. Salvador, P. Chan, Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, in: Proceedings of the Sixteenth IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, IEEE, Piscataway, NJ, USA, 2004, pp. 576–584.

[37] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[38] H. Sidenbladh, Multi-target particle filtering for the probability hypothesis density, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 800–806.
[39] I.R. Goodman, R.P.S. Mahler, H.T. Nguyen, Mathematics of Data Fusion, Kluwer Academic Publishers, Dordrecht, Netherlands, 1997.
[40] C. Mårtenson, P. Svenson, Sensor allocation using equivalence classes of enemy paths, in: Proceedings of the Eighth International Conference on Information Fusion, Philadelphia, PA, USA, International Society of Information Fusion, Sunnyvale, CA, USA, 2005.
[41] P. Svenson, H. Sidenbladh, Determining possible avenues of approach using ANTS, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 1110–1117.
[42] L.A. Klein, Millimeter-Wave and Infrared Multisensor Design and Signal Processing, Artech House Publishers, Norwood, MA, USA, 1997.
[43] J. Johnson, W. Lawson, Performance modeling methods and problems, in: Proceedings of the IRIS Imaging Systems Group, 1974.
[44] MySQL, Online, October 2005. Available from: <http://www.mysql.com>.
[45] P.G. Gonsalves, G.J. Rinkus, S.K. Das, N.T. Ton, A hybrid artificial intelligence architecture for battlefield information fusion, in: Proceedings of the Second International Conference on Information Fusion, Sunnyvale, CA, USA, International Society of Information Fusion, Sunnyvale, CA, USA, 1999, pp. 463–468.
[46] C. Sutton, C. Morrison, P.R. Cohen, J. Moody, J. Adibi, A Bayesian blackboard for information fusion, in: P. Svensson, J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion, June 2004, vol. II, International Society of Information Fusion, Mountain View, CA, USA, 2004.
[47] C.G. Looney, Exploring fusion architecture for a common operational picture, Information Fusion 2 (4) (2001) 251–260.
[48] C.G. Looney, L.R. Liang, Cognitive situation and threat assessments of ground battlespaces, Information Fusion 4 (4) (2003) 297–308.
[49] L.R.M. Johansson, N. Xiong, H.I. Christensen, A game theoretic model for allocation of mobile sensors, in: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia, International Society of Information Fusion, Sunnyvale, CA, USA, 2003, pp. 583–590.
[50] L.R.M. Johansson. Information acquisition in data fusion systems. Licentiate thesis TRITA-NA-03-28, Technical report, Numerical Analysis and Computer Science, Computer Vision and Active Perception Laboratory, Royal Institute of Technology, Stockholm, Sweden, 2003.
[51] R. Suzić, A generic model of tactical plan recognition for threat assessment, in: Proceedings of the SPIE Defense and Security Symposium Orlando, FL, USA, SPIE, Bellingham, WA, USA, 2005.
[52] L.R.M. Johansson, R. Suzić, Bridging the gap between information need and information acquisition, in: P. Svensson, J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion, Mountain View, CA, June 2004, vol. II, International Society of Information Fusion, 2004, pp. 1202–1209.