Approaches to Multisensor Data Fusion in Target Tracking: A Survey

Duncan Smith and Sameer Singh, *Member*, *IEEE*

Abstract—The tracking of objects using distributed multiple sensors is an important field of work in the application areas of autonomous robotics, military applications, and mobile systems. In this survey, we review a number of computationally intelligent methods that are used for developing robust tracking schemes through sensor data fusion. The survey discusses the application of the various algorithms at different layers of the JDL model and highlights the weaknesses and strengths of the approaches in the context of different applications.

Index Terms—Distributed sensors, tracking, information fusion, data fusion.

1 Introduction

Combining the results of multiple sensors can provide more accurate information than using a single sensor [1], [2]; this allows either improved accuracy from existing sensors or the same performance from smaller or cheaper sensors. This paper has been written to complement the landmark survey paper on the subject [3], adding some of the notable breakthroughs of the last decade in fields such as sensor management and distributed sensing. Multi Sensor Data Fusion (MSDF) is used in many diverse fields, although most of the literature addresses the fields of military target tracking [4] or autonomous robotics [5].

Military distributed data fusion is used to facilitate Network Centric Warfare (NCW) [6], [12] or Network Enabled Capability (NEC) [7]. If platforms such as warships and airplanes are networked together and their data is shared, then they will be able to compile a more accurate picture of their environment than with just data from their own sensors. An NEC system contains three vital components [8]:

1. a collection of sensors to generate observations,
2. an automatic processing system to convert data into information and knowledge, and
3. a high-speed communications network to enable the process.

Sensors may be clustered together such as on a submarine, which may have several sonar onboard, or may be carried individually by soldiers [9]. Henceforth, the word "platform" will be used to describe any object that carries sensors. At any fusion processing node, data may therefore come from one of three sources [10] (see Fig. 1):

1. Data type 1: Data from a platform's own sensors, known as "organic data."
2. Data type 2: Network connections to other platforms.
3. Data type 3: A database of data previously received, and of local track estimates.

Traditionally, military data fusion architectures have been centralized or hierarchical [1]. There are, however, many advantages to decentralized schemes, which include lighter processing load, no requirement for a single centralized database, lower communication load, reduced possibility of data flow bottlenecks, and high survivability as there is no longer a single point of failure [11].

To facilitate decentralized fusion, three main issues need to be addressed:

1. Architecture—The way in which nodes connect and share information. For detailed coverage of this aspect of MSDF, see [12], [13], and [14] for a military perspective or [15] for autonomous systems.
2. Sensor management—The way in which sensors are placed to maximize coverage of an area for different tactical goals [16].
3. Algorithms—The way in which processing should be performed.

Although this paper focuses on the military applications of MSDF, it is also readily applicable to robotics. Robots are required to move around autonomously in unknown environments. Due to factors such as cost, reliability, and ease of use, the two most common sensors on this sort of mobile robot are ultra-sonic sonars and digital video cameras [17], [18]. MSDF is required to combine and process the data. This has traditionally been performed by some form of Kalman [19] or Bayesian filter; however, in recent years, there has been a trend toward the use of soft techniques such as fuzzy logic and artificial neural networks (ANNs) [20].

Although more than 30 fusion architectures have been proposed [21], the most widely cited model for data fusion was created by the American Joint Directors of Laboratories Data Fusion Subpanel [22]. This divided the data fusion process into four levels, which make up a hierarchy of

• *D. Smith is with the Research School of Informatics, Loughborough University, Loughborough LE11 3TU, UK, and QinetiQ, Winfrith Technology Centre, Winfrith Newburgh, Dorchester, Dorset DT2 8XJ, UK. E-mail: desmith@qinetiq.com.*
• *S. Singh is with the Research School of Informatics, Loughborough University, Loughborough LE11 3TU, UK. E-mail: s.singh@lboro.ac.uk.*

Manuscript received 27 Feb. 2006; accepted 15 May 2006; published online 18 Oct. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0103-0206.

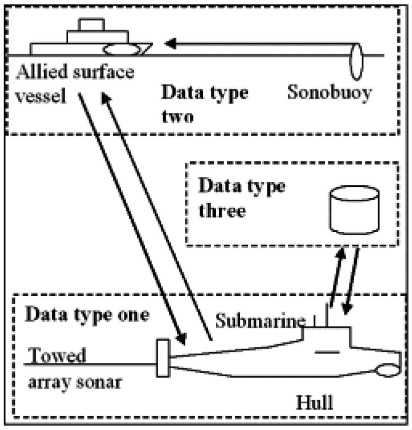

Fig. 1. Three possible data sources for a network enabled submarine.

processing. Although this is by no means the only hierarchy for data fusion and is primarily focused on military applications, it does provide a useful structure with which to classify fusion algorithms. Sections 2, 3, 4, and 5 are divided into the four levels of the JDL model to enable the comparison of similar algorithms.

2 JDL Level 1—"Object Refinement"

Object refinement is usually partitioned into data registration, data association, position attribute estimation, and identification [23]. These four categories and the algorithms that fit within them are outlined in Sections 2.1, 2.2, 2.3, and 2.4. Some algorithms do not directly fit into a single category; for example, [24], [25], [26] all created algorithms which estimate attributes and identification as two complimentary processes by fusing the information from two or more sensors. Association and state estimation has also been performed in a single step [27] to improve performance.

2.1 Data Registration

Data registration functions align the data into a common frame of reference. This is often to change coordinate systems from self-centerd Cartesian coordinates to latitude, longitude, and height above sea level for example.

2.2 Data Association

The association step compares measurements and attempts to collect measurements originating from the same real-world object into a single track. The difficulty is in distinguishing from which target, if any, each measurement originates. This is addressed by measurement-to-track association.

In a distributed system, association can also be the step where tracks from different processing nodes are compared, to combine tracks that are estimating the state of the same real-world object. This is track-to-track association. Sections 2.2.1, 2.2.2, 2.2.3, 2.2.4, and 2.2.5 describe the various approaches for data association.

2.2.1 Nearest Neighbor

Nearest neighbor is the simplest form of association algorithm. In this algorithm, the nearest measurement to the established track is chosen to update the track. This algorithm is very simple and capable of finding a viable solution with very little computational cost. However, in a dense environment, this may lead to many pairings with a similar probability, so errors are typically large [28]. "All neighbor" is another related technique in which all measurements within a gated region are included in the track [28].

2.2.2 Joint Probabilistic Data Association (JPDA)

Bar-Shalom et al. have created two related filters. The first is the Probabilistic Data Association Filter (PDA) [29]. This works for the single-target case only. All measurements in the gated region around a track are assumed to be possible updates for that track. An a posteriori probability of association is calculated for each of the validated measurements. These probabilities are used as weights to calculate a weighted average measurement update, which is added to the track.

The second is JPDA [30], which extends the PDA to the multiple target case. In this, the measurement-to-target association probabilities are computed jointly across all of the targets. For the JPDA to work, every measurement must be assumed to fall within the validation region to ensure that the PDF of all of the false measurements are the same [31].

Bloem and Blom [32] found that the JPDA has no explicit method for track creation, but assumes that the track already exists. Unless specific logic is provided, when new targets appear they simply get absorbed into the old tracks, rather than creating new tracks of their own. Another problem is that all measurements update all targets, which means that if a track is initiated by noise, it will be updated and kept alive by the measurements for other tracks around it, a problem exacerbated by the fact that there is no built-in method for handling expired tracks. Both the PDA and JPDA also suffer from exponential computational complexity.

Due to these problems, [32] developed a novel nearest-neighbor-based approach, the Exact Nearest Neighbor PDA (ENNPDA), and found it was both more computationally efficient and more accurate than the JPDA for scenarios without clutter.

The JPDA's ability to handle clutter was combined with the ENNPDA's ability to avoid tracks merging, known as track coalescence, to create the Coupled PDA (CPDA) [33]. At low target velocities, this algorithm far outperforms a standard JPDA and narrowly outperforms the ENNPDA, while, at higher velocities, the performance of the three algorithms converges [34]. The JPDA has also been extended by [35] to a multisensor JPDA (MSJPDA). This is shown to have superior performance to both the multisensor version of the nearest-neighbor algorithm and to the single sensor JPDA.

2.2.3 Lagrangian Relaxation

With multiple sensors tracking multiple targets, the data association problem can be shown to be NP hard [36]. The evidence in [37] shows that, in all likelihood, an NP-Hard algorithm cannot be solved in a computationally efficient manner, but many approximation algorithms exist to find near optimal solutions. Lagrangian relaxation is one such technique.

Pattipatti et al. [36] were the first to apply this technique to the data association problem. Their solution guarantees polynomial time performance and has memory requirements of only $O(n)^3$. The algorithm consists of two phases: The costs are first assigned to all feasible associations, while, in the second phase, the feasible solution that maximizes the general likelihood ratio is obtained via a 3D assignment algorithm.

2.2.4 Artificial Neural Networks (ANNs)

Track to track data association takes the tracks formed on multiple sensors and attempts to associate or group the tracks that correspond to the same target. With more than two targets, this problem is NP hard and an approximation technique is required to find a solution. Winter and Favier [38] proposed a way of using ANNs to solve this problem. It was shown by [38] that this neural network approach, based upon Hopfield neural networks, always finds a the optimal solution 17.4 percent of the time and found a solution that approximates the true solution the rest of the time.

2.2.5 Fuzzy Logic

The disadvantage of the commonly used PDA and JPDA approaches is that, as the number of targets increases, the amount of computation time rises exponentially. Fuzzy logic algorithms use "common sense" instead of mathematics to find the solution [39].

Hong et al. [40] devised a number of fuzzy rules for data fusion and converted the data into fuzzy sets with the values {NL, NS, ZO, PS, PL} ({negative large, negative small, zero, positive small, and positive large}). The results [40] showed that using fuzzy inference leads to a lower average RMS position error than JPDA and the more fuzzy sets there were, the higher the accuracy. The computational expense of the fuzzy multitarget tracking system is lower than that of the JPDA. Although more fuzzy sets mean higher accuracy, it also increases the computational burden, and, therefore, the appropriate number of sets should be chosen for the desired accuracy.

The Fuzzy Data Association (FDA) algorithm [41] performed data association in a similar way to the JPDA, though the input and output values were encoded in fuzzy sets. When performing radar to infrared fusion, this was found to have a far lower RMS position error than JPDA for both simulated targets used in the experiment. FDA is shown to be less compuationally expensive than JPDA.

In [42], Wang et al. looked at the problem of fusing Electronic Support Measures (ESM) with traditional radar. Above water, radar is the most important sensor as it can provide accurate target location information with both bearing and range from the sensor. ESM is passive, detecting radar signals emitted by targets, and can therefore only produce angular measurements. As a target may have many emitters, it is possible to have multiple ESM tracks fused with a single radar track. In this study, Wang et al. calculated a fuzzy synthetic similarity degree, based upon the residuals between the bearing predicted from the radar measurements and the bearing actually measured on the ESM sensor. Two thresholds were calculated to create fuzzy sets describing the correlation between the ESM and the radar data. For a pairing, there can be:

1. Firm correlation, ESM signals go with radar track.
2. Tentatively correlated, ESM probably goes with radar track.
3. Tentatively uncorrelated, ESM probably do not go with radar track

This enabled the algorithm to determine which ESM tracks to fuse with which radar tracks.

2.3 Position/Attribute Estimation

Position and attribute estimation is the process of taking the associated measurements and calculating the target's state. An example is Target Motion Analysis (TMA) for passive sonar. Passive sonar can only measure the bearing of the target, not the distance. It is necessary to perform TMA to calculate the range and velocity of the target. In Sections 2.3.1, 2.3.2, 2.3.3, 2.3.4, and 2.3.5, we review the most popular methods for position/ attribute estimation.

2.3.1 Kalman Filter (KF)

The Kalman Filter (KF) [43] was first proposed in the 1960s and it is the most commonly used technique in target tracking and robot navigation ever since. The basic KF has been shown to be a form of Bayesian filter [44] that is an optimal estimator for linear Gaussian systems. Given a series of noisy measurements, the KF is capable of estimating the state of the system.

An extension to the KF is the Extended Kalman Filter (EKF) [45]. This enables data such as bearings-only passive sonar data to be used in the KF. Due to the linearization step, the EKF is suboptimal. The EKF is the most popular tool in the literature for sensor fusion in mobile robot navigation.

Both the KF and EKF were originally used on the data from a single sensor. Willner et al. [46] first developed the idea of combining information from local sensors at a central fusion node to form a more accurate global estimate. The drawback of this algorithm was that each local sensor requires the global estimate, which required two-way communication and negates some of the advantages of parallelization.

It has also been proven [47] that, when the KF is used at a central fusion node to fuse the results of multiple local KFs, the results may be improved by feeding the global estimate back to the local filters as their prior state for the next iteration. As the outputs of the local filters are correlated in time, the performance of such a system can be further improved by only outputting every $n\text{th}$ measurement to the global tracker to obtain near optimal performance [48].

An information theoretic view of the KF and EKF has also been suggested [133]. The Information Filter (IF) or inverse covariance filter is a KF that estimates the information state vector, y, defined $y \equiv P^{-1}x$, where x is the traditional state vector and P is its covariance. The covariance of the information state vector is the inverse of the covariance of the state vector, also known as the Fisher Information Matrix or Information Matrix. In this way, the filter estimates the information matrix directly. This form of filter is especially beneficial when the state vector is larger than the measurement vector. Additionally, the situation of having no prior knowledge of the object being tracked can be represented by initializing the information matrix to zero [49]. Doing this in a KF/EKF would require setting elements of the covariance matrix to infinity, which would be impossible.

In cases where the measurement model is highly non linear, even the EKF may diverge. In this situation, the Sigma Point Kalman Filter family of algorithms can be used [50]. Rather than circulating only the mean through the algorithm, SPKFs circulate a collection of precisely selected points around the mean, called sigma points. In using several points, the nonlinearity is more accurately modeled. The use of several points may make this appear similar to a Particle Filter (see Section 2.3.4); however, SPKFs require an order of magnitude of fewer points and are therefore far less computationally expensive. SPKFs include the Unscented Kalman Filter (UKF) [51]. Van der Merwe et al. [52] observed, however, that even UKFs are still limited to Gaussian distributions.

2.3.2 Multiple Model Algorithms

Static (nonswitching) algorithms have been around since the 1960s, though practical algorithms have only been available more recently. If the model used in the filter is different from the actual system dynamics, then the filter will diverge. This may also happen if the system has multiple modes of operation; the filter can only describe one of them. Target tracking falls into this category as the target will generally move in a straight line, but may also have short periods during a maneuver where it changes direction or speed [53].

Switching with the Markov model is easy and it can be more realistic for systems that have time-varying parameters. Two of the most commonly used schemes are the generalized pseudo-Bayesian (GPB) [54], [55] and interacting multiple model (IMM) [56]. Both of these techniques use a bank of filters, though the IMM will require fewer filters. An advantage of these (shared with other MM filters) is that of modularity; the filter used may be a KF or an EKF or even PDA or JPDA [57]. The baseline IMM is the simplest form of hybrid system. Each filter is a standard KF, where each of these KFs represents a different model, such as stopped or moving [58].

IMMs have also been used in state smoothing. Filtering only uses past measurements, whereas fixed lag smoothing delays processing by a fixed number N of updates. Due to the delay, it can use up to N measurements after the time it is processing. Chen and Tugnait [59] developed an IMM-based fixed-lag smoother and showed that its accuracy in terms of mean squared error increased proportionally to the lag. Any form of lag smoothing introduces an inevitable delay between receiving a measurement and calculating a target state estimate. The delay in the measurement is also proportional to the lag; a zero lag smoother is therefore a filter.

The IMM is often used in conjunction with an association technique such as JPDA to form a multiple target tracking system. The resulting algorithm can be used for tracking closely maneuvering targets [60], [61] or targets in clutter [62], [63]. These combine the data association step from the JPDA with the state estimation abilities of the IMM. Hwang et al. used a hybrid IMM JPDA algorithm to track aircraft for air traffic control [64].

2.3.3 Multiple Resolutional Filtering

Data can be viewed at the level of granularity it arrived at or it can be simplified to a lower level of granularity using the wavelet transform. Performing data processing using a combination of different levels of data granularity is called multiple resolution filtering. This technique has been successfully applied to image processing to improve performance. Unfortunately, it is not possible to directly use the techniques developed for processing images to target tracking as image processing is performed as a batch technique, while target tracking and signal processing must give a new estimate whenever new data arrives.

The authors of [65], [66], [67], [68] overcame these problems and applied MRF to target tracking and signal processing by dividing data into blocks. This processed the data by representing it as a tree structure where the top level represents the highest accuracy, the original data [69]. This could be described as a low-pass filter; as data propagates from higher levels of granularity to the lower levels, the high frequency components of the signal are removed.

The algorithm used is a form of IMM [70]. In any multiple model algorithm, it is important to know when the target is maneuvering so that the algorithm can switch from one model to another. Any tracking algorithm may be used at each level nearest neighbor and JPDA have both been shown to work well [71]. As there is far less data available at the lowest level, the algorithm will run faster and computation time is increased. Significant computational effort may be saved by using different algorithms at different levels. If a maneuver is found at the coarsest level, then the next level up is checked to see if the maneuver can be detected there. If it is found, then the decision is propagated up to the next level of processing. The finest level can identify exactly when the maneuver occurred. This division of labor between processing levels has been shown to give improved performance over traditional IMM/KF filters, especially in environments with high background noise levels [72].

Initially, the resolutions had to be reduced by factors of 2 to the power of an integer, i.e., full rate, half rate, quarter rate, etc. This was changed in [73] when a new method was created for decomposing the data to arbitrary resolutions. Hong et al. [74] showed that the same technique could also be used to track dim or quiet targets and the technique was further extended by Fan et al. [75] to both improve accuracy and operate on multiple targets.

2.3.4 Particle Filter (PF)

Unfortunately, other than in simulated experiments, the error rarely is either known or Gaussian, so a method for filtering using arbitrary probability density functions (PDFs) is required. Earlier attempts at improving upon the results of the EKF involved using an IMM. By setting the different models to represent different Gaussian distributions taking a weighted average of the Gaussian results, arbitrary distributions could be modeled. However, this method cannot be applied automatically [76].

A direct approach to modeling the PDF is to divide the search space into a grid, and using the spaces in the grid to represent points in the PDF. Choosing the grid is, however, a nontrivial task and, especially in multidimensional space, a large number of grid points may become necessary.

The particle filter, also known as the Bootstrap, Condensation, or Monte-Carlo filter, was developed to counter this very problem. Rather than having a fixed grid to represent the PDF, these used movable "particles." Early versions of the particle filter used a fixed number of particles, which led to the particles collapsing to a single point and the filter diverging in the same way that a KF does with a poorly described Gaussian [77].

Gordon et al. [78] developed the "bootstrap filter" or Sequential Importance Resampling (SIR) PF. This introduced a resampling step, required to prevent the filter diverging, which removed the particles with the lowest weights at each step and created new particles at points where the weight was the highest. The bootstrap filter was shown to be more accurate than the EKF for tracking in a system with nonlinear measurements, such as bearings only tracking. Since then, several variants of this bootstrap have been developed, such as versions for multitarget tracking [79], [80] and for maneuvering targets using an IMM PF approach [81], [82], [83]. PFs have been shown to be particularly effective in a distributed sensing environment [84]. A thorough description of the different types of PF may be found in [85].

2.3.5 AI Approaches

Sensor fusion with known statistics relies on well-known techniques such as the Kalman Filter or Bayesian statistics. Where there is no specific statistical model of the uncertainty, other techniques, such as rule-based sensor fusion, fuzzy logic, and neural networks, must be used instead.

Rule-based. One of the simplest approaches to multisensor data fusion was proposed by Flynn [86] in which he gave a simple set of heuristic rules that are often used on autonomous mobile robots to fuse the data from two ranging sensors, the first an ultrasonic sensor, and the second a near infrared proximity sensor [87]. Although the rule-set is simple, it is often very effective, and is used as a baseline comparison method for many new AI fusion techniques.

Artificial neural networks (ANNs). Many authors [87], [88], [89], [90], [91], [92] have successfully used neural networks in sensor fusion. A back propagation (BP) network has been used to give navigational abilities comparable to the state-of-the-art [93]. Multilayer networks require a notoriously long training time and alternatives are available to optimize network size. Radial basis function networks (such as those using localized receptive fields (LRF) [94]) train much faster than BP nets because only one layer of weights needs to be modified.

A problem with ANNs is that determining the appropriate number of hidden units can be more of an art than science. Ash [95] proposed a system of dynamic node creation (DNC) which starts with a small network and increases the size one node at a time until the network is large enough to handle the task in hand. DNC was later applied to data fusion by Ghosh and Holmberg [96], who found that given a large number of nodes, backpropagation networks were prone to overtrain very easily, while a network created using a combination of LRF and DNC did not suffer from this problem, although output encoding networks were found to be the most effective network type overall.

Target state estimation has also been performed using neural networks. For example, the Neurally Inspired Contact Estimator (NICE) [97] is a neural network-based target motion analysis (TMA) algorithm. The NICE algorithm has an equivalent accuracy to the Maximum Likelihood Estimator (MLE), but is an order of magnitude faster.

More recently, genetic algorithms (GAs) have been used to design ANNs for data fusion. Abdel-Aty-Zohdy and Ewing [98] used such a technique to develop a data fusion system for an electronic nose.

Fuzzy logic adaptive filter. The KF assumes a priori knowledge of the process and measurement noise covariances. As these are rarely available in most practical systems, these are estimated. This can have a detrimental impact on the performance of the filter and may even promote divergence. Therefore, having an adaptive filter would give better performance than a standard KF if it solved these problems. Escamilla-Ambrosio and Mort [99] proposed the Fuzzy Logic Adaptive KF (FL-AKF). This adjusted the values of Q and R using fuzzy logic to better fit them to the estimated values of covariance. This appears to work well at sensor fault disgnosis, outlier rejection, and where the error changes over time.

Sasiadek and Hartana [101] extended the work with three new techniques: Fuzzy-Logic-based Adaptive C KF (FL-ACKF), Fuzzy-Logic-based Adaptive D KF (FL-ADKF), and Fuzzy-Logic-based Adaptive F KF (FL-AFKF). All of these are based on the FL-AKF. Results show that these techniques are effective in situations where there are heterogeneous sensors, measuring the same parameters, but with different dynamic and noise statistics.

The FL-AKF has been used in autonomous robotics to fuse the positional information obtained from an odometer with the data from onboard sonar [101]. The odometer, which measures how many times the robot's wheels have turned, is prone to drift. The sonar measures the distance to various external objects in the room; this will be accurate, but only available intermittently. Fusion of the two data sources is used to correct the drift of the odometer. Often, this kind of fusion is performed with a KF, but FL-AKF is shown to give more accurate results [102].

2.4 Identification

The identification step classifies the object that the measurements originate from. For the purpose of this paper, it is assumed that the local platform uses the data from its own sensors to produce its own best estimate of the target identity, along with a confidence value for that identity. Once identified locally, it must be fused with remote estimates to form the global solution.

2.4.1 Bayesian Inference

Bayesian Inference (BI) is a technique that uses probabilities to represent degrees of belief. Bayes' theorem can then be used to make subjective estimates of belief. Hall [23] defined a list of problems with Bayesian inference including:

1. difficulty in defining prior likelihoods,
2. complexity when there are many potential hypotheses and many condition dependent events,
3. hypotheses must be mutually exclusive, and
4. the inability to describe uncertainty in decisions.

2.4.2 Dempster-Shafer (D-S) Rule of Combination

There are circumstances in which Bayesian belief no longer applies. Dempster [103] then, later, Shafer [104] generalized the traditional Bayesian belief model to allow explicit representation of uncertainty. This is required to model the situation in which an classification algorithm cannot classify a target or cannot exhaustively list all of the classes to which it could belong.

However, D-S is not without problems. If one classification algorithm identifies a target as type a with 99 percent belief, while another classification algorithm identifies it as type b with 99 percent belief, but both have a 1 percent belief that it is type c, then D-S combines these to output a classification of C with a probability of 100 percent. This is because it is the only nonconflicting output, but the result is counterintuitive.

Jiu et al. [105] suggested that the total probability from conflicting classifications should be averaged across the classifications that made them, which would lead to a more intuitive result in the example given in the previous paragraph, of classifications a and b given just under 50 percent belief each, while c would be given a little over 1 percent. Yu and Yin [106] have also found a solution to this problem in which they incorporate D-S into the structure of parallel decision fusion structure given in [107]. This also overcame the problem described above. D-S has been used in a variety of fusion contexts including landmine detection [108], autonomous robotics [109], and medical systems [110]. For a comprehensive list of D-S and Bayesian-based algorithms, see [111].

2.4.3 Artificial Neural Networks (ANNs)

A neural network is a massive system of parallel-distributed processing elements, connected in a graph topography. Data is not stored separately from the processing as they are intrinsically linked. One of the most difficult problems in ANNs is choosing the most appropriate network topology for the problem. The choice will depend upon the problem characteristics, the characteristics of the likely approach to solving the problem, and the characteristics of the neural networks to be built. There are also several types of learning rules. These are biologically inspired and govern how the network learns.

In one of the earliest examples of using ANNs to fuse multisensor data for identification, [112] used backpropagation and Hopfield neural networks to identify targets. In backpropagation, the data is supplied to the network and the difference between the input and output is calculated. Weights are changed to improve the result. Once the errors have been minimized for all of the data in the training set, the system is ready to use for test data. Hopfield networks have feedback from output to input, giving a dynamic response. They can be unstable, but stability can be ensured by forcing the weight matrix to be symmetric with zeros along its main diagonal. A recurrent network forms an associative memory. Therefore, like human memory, if a part of the memory is supplied, the network will return the full memory. The associative nature of ANNs was utilized to identify targets given a limited amount of information. In the simple examples given, the networks did not make a single mistake in identifying the targets, showing that it is possible to use ANNs to recognize and identify targets.

Neural networks have since been shown [113] to be an extremely simple, easy to apply method and they outperform other fusion techniques at low correlation levels.

2.4.4 Expert Systems

Although neural networks are good general-purpose problem solvers, they are often too cumbersome for the process and one of the main difficulty with these is finding the correct volume of training data. Given too little or too much training data, the system will make the wrong decisions. Kittler [114] suggests that expert systems should be used to make the identification on each platform and a weighted average is taken of the resultant identification decisions.

2.4.5 Voting and Summing Approaches

Voting and summing fusion are two popular, yet very simple distributed classification approaches. In both, a bank of classification algorithms is used; these may be located with the sensor or with the fusion algorithm. In sum fusion, the confidence of each classifier in each hypothesis is summed and the hypothesis with the highest overall result is used. In voting fusion, the hypothesis which was deemed the most likely by the highest number of sensors is chosen. In situations where the distribution of the errors on the data being fused is Gaussian, the sum algorithm outperforms vote, while the opposite is true in systems where the estimation error has a tail distribution [115]. These have been used in a wide variety of contexts such as biometric classification fusion [116], landmine detection [117], and target tracking [118].

2.4.6 Distributed Classification

It is possible to distribute the classification of targets across the nodes at which the target is detected. Most solutions to this problem involve centralized processing of the data to either classify at the central point or fuse the results of the local classifications together. Caruso and Withanawasam [119] created a scheme in which magnetometers could classify based on the magnetic signature of a vehicle. However, this resulted in a high computational load on each of the sensor nodes and required the magnetic signature of the vehicle to be known for every aspect in which the vehicle might be observed. This would be acceptable in the case given in [119] of the vehicles being driven down a road with the sensor mounted beneath the lane of traffic, but there are many detection situations in which this would be too restrictive. Raghavendra et al. [120] gave a method in which sensors exchanged target feature vectors that could then be used to classify a target, but this placed a high load on the networking. Duarte and Hu [121] relied on each sensor classifying the target and then classifications were centrally fused, leaving a large computational effort at the node.

Arora et al. [122] used "influence fields" to form a noncentralized distributed classifier based on inexpensive binary sensors in their incredibly in-depth paper that covers all details of distributed sensing from the type of sensor to use (magnetometer and radar), detection processing, classification, tracking, and time synchronization. Classification is performed by measuring the size and shape of the area covered by the sensors able to detect the target, its influence field. The classification module passes result to tracker, which tracks classifications over time, both localizing and simultaneously reducing false alarms.

A dilemma faced in constructing a distributed classifier is whether to take the data from all nodes and fuse the data to classify, known as *data fusion*, or to classify locally at each node and fuse the classification results, *decision fusion*. Data fusion would usually require more network bandwidth than the latter. Brooks et al. [123] supposed that data fusion would be a superior choice if the information represented by the data was correlated, while decision fusion would be a better choice if the data was uncorrelated. Additionally, [124] demonstrated that decision fusion worked well when the data was fault-free; however, its performance degraded faster than data fusion when measurement error was introduced to the system.

3 JDL Level 2—"Situation Assessment"

Situation assessment (SA) fuses the kinematic and temporal characteristics of the data to create a description of the situation in terms of indications of warnings, plans of action, and inferences about the distribution of forces and information. An SA algorithm will decide whether and in what way an object is or is likely to act in a hostile manner. Unfortunately, most research is on the lower levels of fusion and, therefore, this area is less well understood [125].

Looney and Liang [126] used a series of algorithms for situation assessmen. First, the uniform k-centralized mean (UKCM) algorithm clustered the detected targets into groups. Once these clusters have been formed, it is possible to assess their intent using a fuzzy belief network. The simple rule-set of the fuzzy belief network and the simple experimental scenario show that this kind of technique is capable of making situational assessments, though a more complex belief network would be required to tackle any real problem.

This is also relevant in nonmilitary contexts, such as context aware processing, in which the task is to develop a machine that is able to understand and react appropriately to its environment. Wu et al. [127] looked at multisensor data fusion from an omnidirectional camera and a microphone to detect the focus of attention of attendees at a meeting. In this study, Dempster-Shafer logic was used to combine the processed outputs of the sensors, such as the location of the meeting and who was talking. This was used to improve the estimate of each attendee's focus of attention compared to the output of the individual sensors.

4 JDL Level 3—"Threat Assessment"

The third level of refinement assesses the threat posed by the enemy being tracked. This may also include an assessment of the friendly forces ability to engage the enemy effectively. Fusion levels two and three are often referred to as "information fusion," while level one is "data fusion." Although this distinction is vague, it is useful as the higher levels tend to utilize symbolic rather than numerical reasoning and tend to be more subjective [128]. In human factors research, this is often referred to as "Situational Awareness" (SA).

Level three of the JDL model has received far less attention in the literature than any of the other levels. Initial papers are starting to appear on the subject, though, at present, they are as much about understanding the challenges of the problem as solving it.

Salerno et al. [21] provided the starting point for a framework for information fusion for SA; it also gives an example situation in which automated situational awareness would be of benefit. The paper concludes with a discussion of metrics that could be used to validate SA techniques.

Jakobson et al. [129] looked at the problem of threat assessment using cognitive fusion techniques. This breaks the problem down into three areas:

1. situation awareness, understanding the meaning of multisensor data, recognizing complex time-dependent patterns, and determining threats and other activities that reveal intent,
2. decision awareness, reasoning about situations and understanding the ramifications of suggested actions, and
3. knowledge awareness, learning and improving skills for fusion procedures, and utilizing historic data to create new fusion patterns and situation classes.

The combination of real-time Event Correlation (EC) and Case-Based Reasoning (CBR) is suggested to produce a generic framework to perform threat assessment. When EC recognizes a series of correlated events, CBR can be used to identify the events as a case, where a case adds further meaning to the set of events and infers a possible situation. Jakobson et al. [129] provided a basis for a possible system, but recommend further work must be done before any such system could be used in a real problem domain.

5 JDL Level 4—"Process Assessment"

The process management stage is an ongoing assessment of the other fusion stages to ensure that the data acquisition and fusion is being performed in a way that will give optimal results. This could also improve results by adjusting the parameters in the fusion process, establishing a target priority [2], or moving the sensors to give improved coverage of the search area [16]. The problem of optimal sensor deployments is closely related to both the alarm placement problem, which is known to be NP complete and the Knapsack problem, which is known to be NP complete [130].

Penny [131], [132] found a strategy for locating a submarine as quickly as possible using passive sonobuoy sensors, which was shown to reduce the detection times up to four times. Hernandez et al.. [133] generalized these results to create a framework for the systematic management of multiple sensors in target tracking in the presence of clutter.

Niu et al. [134], [135] gave a method for optimally distributing the sensors in time the results show that if the target has a high probability of detection and a medium or high maneuvering index, then time-staggered sensors (sensors with updates arriving in turn) should be used. In other circumstances, there is little between staggered and synchronized (arriving at the same time) sensor updates. If two sensors have drastically different performances, then optimal results are obtained by keeping them synchronized. If they have similar or identical performance, then they should be staggered uniformly.

MultiSensor Management (MSM) was discussed by [136], who argued that multisensor management affected all levels of the JDL model. They described MSM as a top down approach, which begins at level 4, but continues down right to level 1 as follows:

- Level 4 (mission planning)
  - Which service to perform?
  - Which accuracy level?
  - What area of the environment to focus on?
- Level 3 (resource deployment)
  - What extra sensors are required?
  - Where to place the new sensors?
- Level 2 (resource planning)
  - Sensor selection for multisensor tracking.
  - Sensor cueing, handing tracks from one sensor to another.
- Level 1 (sensor scheduling)
  - Timeline of commands for each individual sensor.

5.1 Distributed Sensing

Process assessment has also been covered in the distributed sensing literature; here, it is a matter of dynamically selecting which sensors to use in order to gain the most information in the most efficient way. The idea of using Shannon information theory for this was first proposed by [137], selecting sensors based on expected information gain was first suggested by [138]. Wang et al. [139] more recently showed a technique for dynamically selecting the sensor to request data from in order to maximize the information gain. Wang et al. [139] used greedy selection of the next sensor; of all of the unused sensors, the one predicted to give the largest information gain is used.

Moore et al. [140] developed a system that made the most of limited resources on distributed nodes by designing a mobile code daemon. This daemon allowed a node to download the classifiers or trackers it required as it found that it needed them, at the same time clearing out the ones that were no longer required. This allowed the system to configure itself dynamically. Friedlander et al. [141] extended [140] to create a system in which nodes form themselves into clusters or coalitions. This avoids the "curse of dimensionality" problem that is troublesome in very large systems. Without this, each node would be forced to share information with every other, meaning that the processing and communications burden increases with each node added to the network, while, in the proposed scheme, nodes only share information with those in the same coalition.

In another coalition forming technique, [142] discussed how to form dynamic coalitions of autonomous nodes. Dynamic coalitions are teams that form to perform a task when a single node would not have sufficient resources to perform the task. Nodes learn how to form coalitions that are more productive. Experimental results show that these cooperative agents can track targets far better than trackers that simply react individually and are able to share computational resources, allowing faster and more efficient processing.

Horling et al. [143] and Yadgar et al. [144] discussed methods of creating hierarchy in which the nodes are divided up geographically into coalitions and each coalition is given a team leader. In [143], each track detected is allocated a track leader by the team leader and this node instructs the other nodes in the coalition. The technique is made to work using a conflict resolution strategy, which is required when nodes are given two conflicting tasks. Yadgar et al. [144] investigated how the number of levels in an architecture may affect performance and found that, as the number of levels in the hierarchy increases, the number of targets it is capable of tracking decreases; however, the amount of time required by an individual node to complete its mission decreases exponentially.

Ortiz et al. [145] proposed an auction-based technique called Dynamic Mediation (DM) for forming and allocating work to cooperating teams of nodes. In DM, the bid is not simply an individual value bid from a particular node, but a bid from a team of nodes, which can include information such as positive or negative interactions with other jobs allocated to the team. Experimental results from [145] suggest that DM shows the largest performance improvement over a traditional auction where time is limited.

Liu et al. [84] resolved the curse of dimensionality by separating the processes of allocating data points to targets being tracked and position estimation. Targets far away from each other are tracked separately in the traditional way, while targets close together are tracked jointly. As more than one target may be tracked at the same time, the PDF will not be Gaussian. This led [84] to use a particle filter as a position estimation algorithm as it can estimate arbitrary distributions.

Akyildiz et al. have written a comprehensive survey [146] on the subject from a networking perspective. This gives a description of the major network topologies and protocols available and concludes that there remain many unsolved problems in sensor network research such as fault tolerance, scalability, node cost, and power consumption. In another survey paper [147], the same authors outline the major applications for sensor networks, citing examples such as:

- military applications, such as monitoring friendly forces and battle damage assessment,
- environmental applications, such as bird migration monitoring, or flood detection,
- health applications, such as tracking doctors within a hospital, or remotely monitoring a patient's physiological data, and
- home automation.

6 Challenges for Multisensor Tracking

There are two main challenges for distributed multisensor-based tracking: 1) The order in which data arrives may not be suitable for processing and measurements that arrive out of sequence and 2) the effect of one sensor on another or data correlation.

6.1 Out of Sequence Measurements (OOSMs)

Each measurement arrives at the fusion algorithm with a discrete "timestamp." In a multisensor tracking system, there will be various propagation times from the different data sources, it is clearly possible (in most multisensor systems, probable) that some data will arrive out of sequence. Typically, fusion systems only retain a history of the most important statistics, such as the state estimate and the covariance matrix. Therefore, the problem is to find a way to update the current estimate using the OOSM.

OOSMs have been shown to be a problem as far back as the 1980s [148]. The early attempts to solve the problem of delayed measurements relied upon no newer measurements being received in between the time of the measurement being created and the time at which it is processed [149], [150]. These worked well, but only solved the problem of the data being delayed, not the fact that the data would often be in the wrong order.

An intuitive and perfect solution in terms of accuracy is to store all input in order of time, then, when an OOSM is received, reprocess all data. Although this gives the optimal solution, it is prohibitively intensive in terms of both computation and memory requirements.

Blackman and Popoli [151] and Hilton et al. [152] proposed an approximate solution to the problem called "Algorithm B." Bar-Shalom [153] later extended this to create an algorithm with optimal output, "Algorithm A." All of these, however, assumed that the lag in the OOSM was less than one time step. Bar-Shalom [153] also showed that Algorithm B is nearly optimal.

Since these papers were written, there have been many attempts to extend the algorithms to work for arbitrary lags [154], [155], [156]. Bar-Shalom et al. [157], [158] not only showed that these were more expensive in terms of both computational load and memory usage, but also proposed a new way to utilize the OOSM by processing in a single step, calculations that are very similar to a standard Kalman Filter KF. Adjustments are made for both the A and B algorithms. The new arbitrary lag algorithms are called "Al" and "Bl," respectively. Although the results from Al are optimal, the processing time required is very high. Bl is recommended as it considerably faster, with comparable results to algorithm A.

Hong et al. have recently extended the multirate IMM (MRIMM) algorithm [70] to work on the OOSM problem [159], [160], [161]. The nature of multirate filtering lends itself well to the incorporation of OOSMs as it provides both an efficient processing structure for state estimate retrodiction and an efficient memory structure for storing historical information. Other algorithms, such as the IMM [162] and multiple hypothesis tracking [163] have also been extended to enable tracking in clutter or with maneuvering targets while processing OOSMs.

As all of these techniques are KF-based, they do not help when using a particle filter (PF), as to use them would be impossible due to the necessary computational load. Earlier attempts at solving the OOSM problem concentrated on regenerating the PDF at the time of the OOSM [153]. In a PF, this would require a large amount of computational resources. To avoid this problem, Orton and Marrs [164] stored the distribution of particles at each time step to avoid having to recalculate them. It was later shown in this technique that, as delays increase, the result only worsens slightly and is still very close to optimal [165]. Unfortunately, [166] showed that this required an infeasible amount of storage to keep the state of every data point for every time update and described an efficient method for retrodicting the state in the same way as the KF-based approaches.

It is worth noting that [166] described the PF-based algorithm that performs similarly, though slightly worse than the equivalent EKF-based solution. Mallick and Marrs wrote a comprehensive comparison of KF and PF-based OOSM filters [167], which found that, for linear measurement models, the OOSMKF algorithm produced optimal results. The numerical results suggested that the OOSMPF algorithm was suboptimal. Experiments with nonlinear GMTI radar data show that the bias errors in the state estimates from the OOSMKF and OOSMPF were small and comparable. The OOSMPF is shown to be comparable to the OOSMKF for small lags and small values of process noise.

OOSMs are not only a problem for measurement fusion. Challa et al. [168] showed that they would also affect track-to-track fusion and developed an algorithm based on an augmented state KF (AS-KF). This is a KF which processes not only the current state, but also previous states simultaneously. When a track estimate arrived, the measurements that went into calculating it were derived and used to update the algorithm, allowing sequence tracks (OOSTs) to be processed. Challa et al. [168] demonstrated that this gave improved performance over ignoring the OOSTs. To compliment the AS-KF, an augmented state probabilistic data association (AS-PDA) algorithm was also created by [169] to process OOSMs in clutter.

6.2 Data Correlation

One problem with the KF is that it requires either that the measurements are independent or that the cross-covariance is known. A common simplification is to assume the cross-covariance to be zero, though, in this situation, the KF produces nonconservative covariances. This leads to an artificially high confidence value, which can lead to filter divergence [170]. The optimal KF-based approach is the one in which the KF maintains cross covariance information between updates [171], [172], [173], [174], [175]. However, this solution scales quadratically with the number of updates, making it impractical [176]. Taking into account the correct cross correlation does not significantly improve the results, even though it does reduce the false association rate [177].

A common cause of correlation between tracks in a distributed multisensor environment is data incest or rumor propagation. Data incest[1] is the situation in which raw measurements are inadvertently used multiple times. This is caused either by the same information taking several different paths from the other sensor to the fusion node or by cyclic paths in which the information recirculates from output of a fusion node back to the input [178], see Fig. 2. The examples shown in Fig. 2 are very simple and as the

1. Often referred to in the literature simply as correlation.

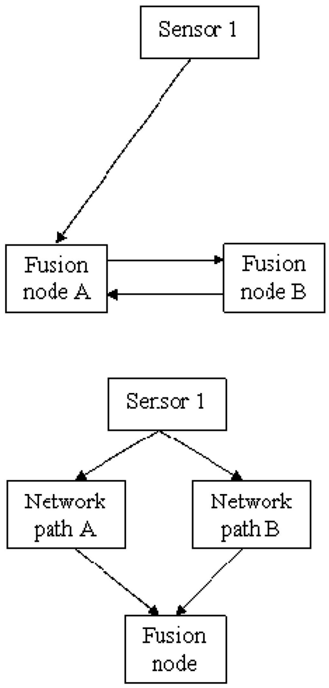

Fig. 2. Data flows that will cause data incest.

number of sensors and the number of measurements increases, data incest becomes harder to identify and cure.

Data incest may at first appear trivial to avoid, however, for fusion, the simplest approach to communication is to send all raw measurements. This requires no processing at the source and relatively straightforward processing at the destination. As long as a record was kept of which track updates had been used, it would be possible to avoid incest. This solution has unfortunately been shown not to be practical as it does not scale well [179], and is therefore unappealing.

Incest is usually studied in terms of the bias caused to the state estimate in localization algorithms. However, it will also adversely affect data fusion for identification. Higher up the JDL model in level 4, incest may have more subtle implications. For example, if a sensor is deployed in an area to confirm a report of a target, then the prior probability of finding such an object must be altered accordingly, or the confidence in identification when a matching target is discovered.

Measurement reconstruction [180], [181] is a technique that can be used in a global fusion node. This compares remote estimates received with its own version of the global estimate. It is then possible to recreate the measurement that caused the estimate to change to the estimate. This way the remote measurements can be recovered, and used in local fusion algorithms. This technique has also been extended [182] to include target tracking in clutter by incorporating JPDA into the technique, then further to include multiple sensors [183] and non-Gaussian error distributions with Gaussian mixture models [184].

McLaughlin et al. [185], [186] developed a data incest removal strategy for distributed architectures. The algorithm takes the state estimates from other nodes and resolves the remote measurements from them. It stores these remote measurements and uses them to update its

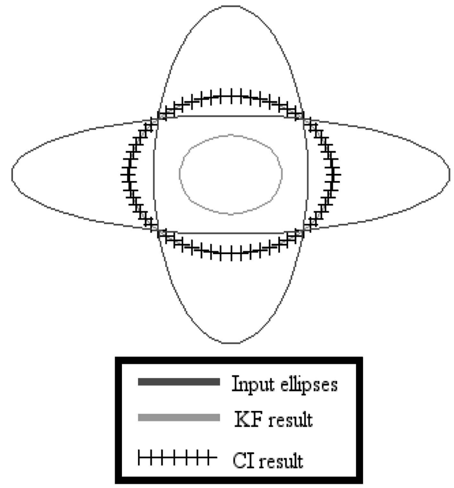

Fig. 3. Covariance intersection.

own state estimate; this way the incest has been removed before the data is fused.

As estimating the cross-covariance is computationally expensive, Julier and Uhlmann proposed the Covariance Intersection (CI) algorithm [187]. This is based upon a simple premise that if the covariances of the two estimates being combined were to be visualized as overlapping ellipses, then the desired resultant covariance would be the smallest ellipse that would surround the intersection of the two ellipses (see Fig. 3). This way the same data may be presented to the filter several times and the covariance will not be reduced as fresh information is not being provided. Julier and Uhlmann [187] also showed that the CI algorithm can work in the case of a networked group of estimators where the network forms a ring. This is important as [188] had shown this to be impossible with a KF.

The use of CI and the KF was also compared by Arambel et al. [189], who used the two algorithms to keep five spacecraft orbiting in strict formation to perform interferometry. Four will reflect light to the fifth, which will turn the measurements into interferometric detection. To do this, the formation will need to be accurate within 0.1m and 0.3 milli radian. The spacecrafts estimate each other's position and pass that information to the other craft in a ring network. This leads to cross correlation between the values being fused. Each craft runs a set of Kalman filters to estimate the state of the entire network. The data can be updated either with readings from sensors or with the estimate of all positions given by the previous spacecraft in the ring. The solution chosen was to use the Kalman filter for data that is received from the organic sensors, and the Covariance Intersection (CI) algorithm for the inorganic measurements. This prevented cross-correlation of data and filter divergence.

Hurley [190] gave an information theoretic proof of the CI technique and pointed out that CI is capable of fusing any probability density function, not just Gaussian distributions. It also states that, although CI is excellent for fusion of densities, if measurement fusion is required, then more traditional fusion techniques are probably more suitable.

Covariance intersection works by taking a weighted average of the two covariances being combined. This weighting is a single value ω, which is used to combine covariances P_A and P_B as follows: $P_C^{-1} = \omega P_A^{-1} + (1-\omega) P_B^{-1}$. Chen et al. [191] showed that, when the covariances being combined are N-dimensional vectors, the weights could be N-dimensional vectors, while CI only searches a one-dimensional curve for possible values. It was found, however, [192] that CI finds the optimal value. This provided formal proof of the optimality of the covariance intersection problem at finding the upper bound for the combined covariance. It also showed that, as CI performs N-dimensional optimization while only searching one-dimensional space, it is a very efficient algorithm.

Finding the weighting value ω is the most computationally complex process in CI. To reduce the computational burden of the algorithm, [193] proposed a suboptimal noniterative algorithm to find ω.

CI works to solve the problem of correlated inputs, but it is undefined for inconsistent inputs. To solve this, Uhlmann developed Covariance Union (CU) [194]. Inconsistent inputs can be detected by calculating the Mahalinobis Distance (MD) between the inputs. If the MD exceeds the threshold, then the union, rather than the intersection of the covariances will be used. Uhlmann [194] also showed that this technique can also reject outliers.

CI is pessimistic, with the ellipse being larger than it needs to be; this is the exact opposite of the EKF. The largest ellipsoid algorithm [195] avoids this by creating the largest ellipse that will fit within the intersection of the covariances (Fig. 4). This is always more optimistic than the CI algorithm.

The Kalman filter is optimal providing the data is from independent sources. If correlation information is missing or incomplete, then the result will be an inconsistent estimate. Covariance intersection avoids this, but its conservative estimates reduce performance. Largest ellipsoid leads to tighter estimates since it underestimates the covariance rather than overestimating it, though this is less of an underestimate than the KF, so filter divergence is still avoided.

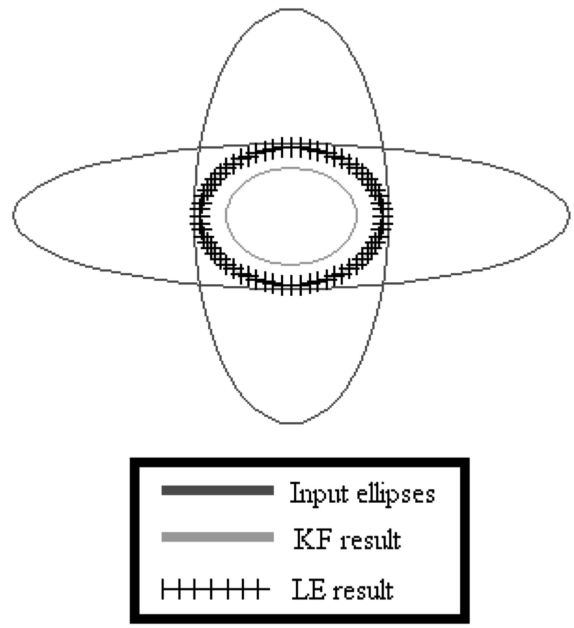

Fig. 4. Largest ellipsoid.

7 Conclusions

A brief overview of contemporary techniques in distributed data fusion was presented in this paper. This discussion is based around the well-known JDL data fusion framework. Multisensor data fusion is shown to be an active area of research, spanning many traditional research areas with applications for industrial control, autonomous robotics, and military tracking.

Much work has focused on the problems associated with the first level of the JDL framework. Level one, "Object Refinement," includes data alignment, data association, position attribute estimation, and identification. The other three levels are less well covered in the literature, possibly as the higher levels require the foundation of the first. Now that this firm theoretical foundation has been laid, work has begun in this area, but it will take many years to catch up with the breadth of work on level one.

In addition to outlining current techniques, a selection of the remaining challenges in data fusion was also discussed. Most of the level 1 data fusion research carried out over the last decade has focused on the problems being introduced by using multiple distributed sensors and, although much has been written on the subject, many problems are yet to be fully solved. For example, many solutions exist for the out-of-sequence measurements (OOSM) problem, but almost all concentrate on measurements that are out of sequence by a few scans at most. A method of accurately utilizing measurements of an arbitrary age is still elusive. Covariance intersection has often been described as a panacea for all cross-correlation problems in data fusion. However, it has yet to solve the problem in the particle filter or in target identity fusion and also provides very pessimistic results for the combined data. We expect that future research will address some of these challenges and develop better methods of uncertainty management.

References

[1] C.-Y. Chong, S. Mori, K.-C. Chang, and W.H. Barker, "Architectures and Algorithms for Track Association and Fusion," *IEEE Aerospace and Electronic Systems*, vol. 15, no. 1, pp. 5-13, Jan. 2000.
[2] E. Waltz and J. Llinas, *Multisensor Data Fusion.* Artech House, 1990.
[3] D.L. Hall and J. Llinas, "An Introduction to Multisensor Fusion," *Proc. IEEE*, vol. 85, no. 1, Jan. 1997.
[4] R.C. Luo, "Multisensor Fusion and Integration—Approaches, Applications, and Future Research Directions," *IEEE Sensors J.*, vol. 2, no 2, Apr. 2002.
[5] R.C. Luo, C.C. Yih, and K.L. Su, "Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions," *IEEE Sensors J.*, vol. 2, no. 2, pp. 107-119, Apr. 2002.
[6] M.G. Ceruti, "Data Management Challenges and Development for Military Information Systems," *IEEE Trans. Knowledge and Data Eng.*, vol. 15, no. 5, pp. 1059-1068, Sept./Oct. 2003.
[7] The Secretary of State for Defence, "Delivering Security in a Changing World,"defence white paper, U.K. Ministry of Defence, Dec. 2003.
[8] N.J. Hammond, "Revolution in Military Affairs—Does Hydrography Have a Part to Play?" *Proc. US Hydrographic Conf.*, 2001.
[9] F. Stein, J. Garska, and P. McIndoo, "Network-Centric Warfare: Impact on Army Operations," *Proc. IEEE/AFCEA EUROCOMM Information Systems for Enhanced Public Safety and Security*, pp. 288-295, 2000.

[10] J. Llinas, "Studying the Complexities in Distributed Object Tracking Systems," *Proc. 2003 IEEE Int'l Conf. Systems, Man and Cybernetics,* vol. 2, pp. 2035-2041, 2003.

[11] M.E. Liggins II, C.-Y. Chong, I. Kadar, M.G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking," *Proc. IEEE,* vol. 85, no. 1, Jan. 1997.

[12] S.A. Moody, "Challenges in Building Scalable Network Centric Real Time Information Dissemination Systems," *Proc. Eighth IEEE Workshop Object Oriented Real Time Dependable Systems,* 2003.

[13] M. Kokar and K.H. Kim, "Review of Multisensor Data Fusion Architectures and Techniques," *Proc. 1993 IEEE Int'l Symp. Intelligent Control,* pp. 261-266, 1993.

[14] A. Gad and M. Farooq, "Data Fusion Architecture for Maritime Surveillance," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 1, pp. 448-455, 2002.

[15] H. Xinhan and W. Min, "Multi-Sensor Data Fusion Structures in Autonomous Systems: A Review," *Proc. 2003 IEEE Int'l Symp. Intelligent Control,* pp. 817-821, 2003.

[16] N. Xiong and P. Svensson, "Multi Sensor Management for Information Fusion: Issues and Approaches," *Elsevier Information Fusion 3,* pp. 163-186, 2000.

[17] G. Benet, J.E. Simo, and M. Martinez, "A Multisensor Robot Distributed Architecture," *Proc. Symp. Information Control in Manufacturing,* vol. 2, pp. 583-588, 1999.

[18] G. De Cubber, H. Sahli, and F. Decroos, "Sensor Integration on a Mobile Robot," *ISMCR 2002: Proc. 12th Int'l Symp. Measurement and Control in Robotics,* June 2002.

[19] J.B. Gao and C.J. Harris, "Some Remarks on Kalman Filters for the Multisensor Fusion," *Elsevier Information Fusion 3,* pp. 191-201, 2002.

[20] R.R. Murphy, "Sensor Fusion," *Handbook of Brain Theory and Neural Networks,* Bradford Book, 2003.

[21] J. Salerno, M. Hinman, and D. Boulware, "Building a Framework for Situational Awareness," *Proc. Seventh Int'l Conf. Information Fusion,* pp. 219-226, 2004.

[22] US Dept. of Defense, Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, *Data Fusion Lexicon,* 1991.

[23] D. Hall, *Mathematical Techniques in Multisensor Data Fusion.* Artech House, 1992.

[24] M.L. Krieg, "Joint Multi-Sensor Kinematic and Attribute Tracking Using Bayesian Belief Networks," *Proc. Sixth Int'l Conf. Information Fusion,* vol. 1, pp. 17-24, 2003.

[25] A. Farian, P. Lombardo, and M. Marselia, "Joint Tracking and Identification Algorithms for Multisensor Data," *IEE Proc. Radar and Sonar Navigation,* vol. 149, no. 6, Dec. 2002.

[26] D.W. Pace, M. Mallick, and W. Eldredge, "Spectral Feature-Aided Multi-Target Multi-Sensor Passive Sonar Tracking," *Proc. IEEE Oceanic Eng. Soc. Conf.,* vol. 4, pp. 2120-2126, 2003.

[27] L. Kester and A. Theil, "Fusion of Radar and EO Sensors for Surveillance," *Proc. Third Int'l Conf. Information Fusion,* vol. 1, pp. TUD1/3-TUD1/9, July 2000.

[28] S.S. Blackman, "Association and Fusion of Multiple Sensor Data," *Multitarget Multisensor Tracking: Advanced Applications,* Y. Bar-Shalom, ed., pp. 187-218, Mass.: Artech House, 1990.

[29] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association," *Automatica,* vol. 11, pp. 451-460, 1975.

[30] Y. Bar-Shalom, "Extension of the Probabilistic Data Association Filter to Multitarget Tracking," *Proc. Fifth Symp. Nonlinear Estimation,* pp. 16-21, Sept. 1974.

[31] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, 1995.

[32] R.J. Fitzgerald, "Development of Practical PDA Logic for Multitarget Tracking by Microprocessor," *Multitarget Multisensor Tracking: Advanced Applications,* Y. Bar-Shalom, ed., 1989.

[33] E.A. Bloem and H.A. P. Blom, "Joint Probabilistic Data Association Methods Avoiding Track Coalescence," *Proc. 34th IEEE Conf. Decision and Control,* vol. 3, pp. 2752-2757, 1995.

[34] H.A.P. Blom and E.A. Bloem, "Probabilistic Data Association Avoiding Track Coalescence," *IEEE Trans. Automatic Control,* vol. 45, no. 2, Feb. 2000.

[35] S.D. O'Neil and L.Y. Pao, "Multisensor Fusion Algorithms for Tracking," *Proc. 1993 Am. Control Conf.,* 1993.

[36] K.R. Pattipatti, S. Deb, Y. Bar-Shalom, and R.B. Washburn, "A New Relaxation Algorithm and Passive Sensor Data Association," *IEEE Trans. Automatic Control,* vol. 37, pp. 198-213, Feb. 1992.

[37] N.J. Nilsson, *Principles of Artificial Intelligence.* Palo Alto, Calif.: Tioga, 1980.

[38] M. Winter and G. Favier, "A Neural Network for Data Association," *Proc. 1999 IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 2, pp. 1041-1044, Mar. 1999.

[39] A.M. Aziz, M. Tummala, and R. Cristi, "Fuzzy Logic Data Correlation Approach in Multisensor-Multitarget Tracking Systems," *Signal Processing 76,* pp. 195-209, Elsevier, 1999.

[40] H. Hong, H. Chong-zhao, Z. Hong-Yan, and W. Rong, "Multi-Target Tracking Based on Multi-Sensor Information Fusion with Fuzzy Inference," *Control and Decision,* vol. 19, no. 3, pp. 272-276, 2004.

[41] Y.M. Chen and H.C. Huang, "Fuzzy Logic Approach to Multisensor Data Association," *Math. and Computers in Simulation 52,* pp. 399-412, Elsevier, 2000.

[42] G.H. Wang, S.Y. Mao, and Y.J. Liu, "Triple Threshold Radar-to-ESM Correlation Algorithm when Each Radar Track Is Specified by a Different Number of Measurements," *IEE Proc. Radar Sonar Navigation,* vol. 147, no. 4, Aug. 2000.

[43] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Eng.,* no. 82, pp. 34-45, 1960.

[44] Y.C. Ho and R.C.K. Lee, "A Bayesian Approach to Problems in Stochastic Estimation and Control," *IEEE Trans. Automatic Control,* 1964.

[45] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles Techniques and Software.* Artech House, 1993.

[46] D. Willner, C.B. Chang, and K.P. Dunn, "Kalman Filter Algorithm for a Multisensor System," *Proc 15th IEEE Conf. Decision Control,* 1976.

[47] J.B. Gao and C.J. Harris, "Some Remarks on Kalman Filters for the Multisensor Fusion," *Elsevier Information Fusion,* vol. 3, pp. 191-201, 2001.

[48] R. Lobbia and M. Kent, "Data Fusion of Decentralized Local Tracker Outputs," *IEEE Trans. Aerospace and Electronic Systems,* vol. 30, no. 3, July 1994.

[49] P.S. Maybeck, "Stochastic Models, Estimation and Control, Volume 1," *Math. in Science and Eng.,* vol. 141-1, Academic Press, 1979.

[50] R. van der Merwe, E.A. Wan, and S. Julier, "Sigma-Point Kalman Filters Nonlinear Estimation and Sensor Fusion—Applications in Integrated Navigation," *Proc. AIAA Guidance Navigation and Controls Conf.,* Mar. 2004.

[51] S.J. Julier and J.K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. AeroSense: The 11th Int'l Symp. Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II,* 1997.

[52] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The Unscented Particle Filter," *Advances in Neural Information Processing Systems (NIPS13),* T.K. Leen, T.G. Dietterich, and V. Tresp, eds., MIT Press, Dec. 2000.

[53] S. McGinnity and G.W. Irwin, "Multiple Model Bootstrap Filter for Manoeuvring Target Tracking," *IEEE Trans. Aerospace and Electronic Systems,* vol. 36, no. 3, July 2000.

[54] G.A. Ackerson and K.S. Fu, "On State Estimation in Switching Environments," *IEEE Trans. Automatic Control,* vol. 5, no. 1, 1970.

[55] J.M. Bruckner, H.R.W. Scott, and G.R. Rea, "Analysis of Multimodal Systems," *IEEE Trans. Aerospace and Electronic Systems,* vol. 9, no. 6, pp. 883-888, 1973,

[56] Y. Bar-Shalom and H.A.P. Blom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Trans. Automatic Control,* vol. 33, no. 8, Aug. 1988.

[57] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting Multiple Model Methods in Tracking: A Survey," *IEEE Trans. Aerospace and Electronic Systems,* vol. 34, no. 1, Jan. 1998.

[58] S. Coraluppi and C. Carthel, "Recursive Track Fusion for Multi-Sensor Surveillance," *Information Fusion 5,* pp. 23-33, 2004.

[59] B. Chen and J.K. Tugnait, "An Interacting Multiple Model Fixed Lag Smoothing Algorithm for Markovian Switching Ssystems," *Proc. 37th IEEE Conf. Decision and Control,* Dec. 1998.

[60] M. Hadzagic, H. Michalska, and A. Jouan, "IMM-JVC and IMM-JPDA for Closely Maneuvering Targets," *Conf. Record 35 Asilomar Conf. Signals, Systems, and Computers,* vol. 2, pp. 1278-1282 Nov. 2001.

[61] H.A.P. Blom and E.A. Bloem, "Interacting Multiple Model Joint Probabilistic Data Association Avoiding Track Coalescence," *Proc. 41st IEEE Conf. Decision and Control,* Dec. 2002.

[62] L. Hong, "An Interacting Multipattern Probabilistic Data Association (IMP-PDA) Algorithm for Target Tracking," *IEEE Trans. Automatic Control,* vol. 46, no. 8, Aug. 2001.

[63] H.A.P. Blom and E.A. Bloem, "Combining IMM and JPDA for Tracking Multiple Maneuvering Targets in Clutter," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 1, pp. 705-712, 2002.

[64] I. Hwang, H. Balakrishnan, K. Roy, and C. Tomlin, "Multiple-Target Tracking and Identity Management in Clutter, with Application to Aircraft Tracking," *Proc. 2004 Control Conf.* vol. 4, pp. 705-712, 2004.

[65] L. Hong, "Optimal Multiresolutional Distributed Filtering," *Proc. 31st Conf. Decision and Control,* Dec. 1992.

[66] L. Hong and T. Scaggs, "Real Time Optimal Multiresolutional Sensor/Data Fusion," *Proc. 1993 IEEE Int'l Conf. Robotics and Automation,* vol. 2, pp. 117-122, May 1993.

[67] L. Hong, "Multiresolutional Tracking Using the Wavelet Transform," *Proc. 32nd Conf. Decision and Control,* Dec. 1993.

[68] L. Hong, "Multiresolutional Multiple-Model Target Tracking," *IEEE Trans. Aerospace and Electronic Systems,* vol. 30, no. 2, Apr. 1994.

[69] L. Hong, "Multiresolutional Filtering Using the Wavelet Transform," *IEEE Trans. Aerospace and Electronic Systems,* vol. 29, no. 4, Oct. 1993.

[70] L. Hong, "Multirate Interacting Multiple Model Filtering for Target Tracking Using Multirate Models," *IEEE Trans. Automatic Control,* vol. 44, no. 7, pp. 1326-1340, July 1999.

[71] L. Hong, "Two Level JPDA-NN and NN-JPDA Tracking Algorithms," *Proc. Am. Control Conf.,* 1994.

[72] L. Hong, "Multiresolutional Distributed Filtering," *IEEE Trans. Automatic Control,* vol. 39, no. 4, Apr. 1994.

[73] L. Hong, "Fusing Multiresolutional Data Using Filter Banks," *Proc. Am. Control Conf.,* vol. 2, pp. 1309-1313, June 1994.

[74] L. Hong, J.R. Werthmann, G.S. Bierman, and R.A. Wood, "A Multiresolutional Approach to Target Tracking," *Proc. IEEE 1993 Nat'l Aerospace and Electronics Conf.,* vol. 1, pp. 388-392, 1993.

[75] T. Fan, C.Y. Yang, S.Y. Mao, and S.H. Li, "Multi-Resolution Multiple-Model Target Tracking Based on Model Mixing," *Proc. IEEE Int'l Radar Conf.,* 2000.

[76] I. Errington, "Non-Linear Estimation in Bearings Only Tracking of Manoeuvring Targets," *Proc. IEE Colloqium on State Estimation in Aerospace and Tracking Applications,* Dec. 1989.

[77] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved Particle Filter for Non-Linear Problems," *IEE Proc.—Radar, Sonar, and Navigation,* vol. 146, no. 1, pp. 2-7, 1999.

[78] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel Approach to Non-Linear/Non-Gaussian Bayesian State Estimation," *IEE Proc.-F,* vol. 140, no. 2, Apr. 1993.

[79] C. Hue and J.P. Le Cadre, "Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion," *IEEE Trans. Signal Processing,* vol. 50, no. 2, Feb. 2002.

[80] G.V. Veres and J.P. Norton, "Improved Particle Filter for Multi-target-Multisensor Tracking with Unresolved Applications," *IEE Target Tracking: Algorithms and Applications,* vol. 1, pp. 12/1-12/5, 2001.

[81] S. McGinnity and G.W. Irwin, "Multiple Model Bootstrap Filter for Manoeuvring Target Tracking," *IEEE Trans. Aerospace and Electronic Systems,* vol. 36, no. 3, July 2000.

[82] H.A.P. Blom and E.A. Bloem, "Tracking Multiple Manouvring Targets by Joint Combinations of IMM and PDA," *Proc. 42nd IEEE Conf. Decision and Control,* Dec. 2003.

[83] H.A.P. Blom and E.A. Bloem, "Joint IMMPDA Particle Filter," *Proc. Sixth Int'l Conf. Information Fusion,* vol. 2, pp. 785-792, 2003.

[84] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed State Representation for Tracking Problems in Sensor Networks," *Proc. Third Int'l Symp. Information Processing in Sensor Networks,* pp. 234-242, Apr. 2004.

[85] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, Feb. 2002.

[86] A.M. Flynn, "Combining Sonar and Infrared Sensors for Mobile Robot Navigation," *Int'l J. Robotics Research,* pp. 5-14, Dec. 1988.

[87] H. Martínez Barberá, A. Gómez Skarmeta, M. Zamora Izquierdo, and J. Botiá Blaya, "Neural Networks for Sonar and Infrared Sensors Fusion" *Proc. Third Int'l Conf. Information Fusion,* vol. 2, pp. WEB4/18-WEB4/25, 2000.

[88] M. Buchberger, K.W. Jorg, and E. von Puttkamer, "Laser Radar and Sonar Based World Modelling and Motion Control for Fast Obstacle Avoidance of the Autonomous Mobile Robot MOBOT-IV," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 534-539, 1993.

[89] C.C. Chang and K.T. Song, "Ultrasonic Sensor Data Integration and Its Application to Environment Perception," *J. Robotic Systems,* vol. 13, no. 10, pp. 663-677, 1996.

[90] J.D. Courtney and A.K. Jain, "Neural Network Learning of Variable Grid-Based Maps for the Autonomous Navigation of Robots," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 40-45, 1994.

[91] V. Santos, J. Gonzalves, and F. Vaz, "Perception Maps for the Local Navigation of a Mobile Robot: A Neural Network Approach," *IEEE Int'l Conf. Robotics and Automation,* pp. 2193-2198, 1994.

[92] S. Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence,* vol. 99, no. 1, pp. 21-71, 1998.

[93] D.A. Pomerleau, *ALVINN: An Autonomous Land Vehicle in a Neural Network, Advances in Neural information processing Systems I,* D.S. Touretzky, eds., Morgan Kaufmann, 1988.

[94] J. Moody and C. Darken, "Fast Learning in Networks of Locally Tuned Processing Units," *Neural Computation,* vol. 1, pp. 281-294, 1989.

[95] T. Ash, "Dynamic Node Creation in Backpropagation Networks," technical report, Inst. for Cognitive Science, Univ. of California, San Diego, 1989.

[96] J. Ghosh and R.L. Holmberg, "Multisensor Fusion Using Neural Networks," *Proc. Second IEEE Symp. Parallel and Distributed Processing,* 1990.

[97] C.M. DeAngelis and J.E. Whitney, "The Neurally Inspired Contact Estimator (NICE)," *Proc. IEEE Oceanic Eng. Soc. Conf.,* vol. 3, pp. 1619-1623, 1998.

[98] H.S. Abdel-Aty-Zohdy and R.L. Ewing, "Intelligent Information Processing Using Neural Networks and Genetic Algorithms," *Proc. 43rd Midwest Symp. Circuits and Systems,* pp. 840-845, Aug. 2000.

[99] P.J. Escamilla-Ambrosio and N. Mort, "Multisensor Data Fusion Architecture Based on Adaptive Kalman Filters and Fuzzy Logic Performance Assessment," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 2, pp. 1542-1549, 2002.

[100] P. Jorge Escamilla-Ambrosio and N. Mort, "Hybrid Kalman Filter—Fuzzy Logic Adaptive Multisensor Data Fusion Architectures," *Proc. 42nd IEEE Conf. Decision and Control,* Dec. 2003.

[101] J.Z. Sasiadek and P. Hartana, "Sensor Data Fusion Using Kalman Filter," *Proc. Third Int'l Conf. Information Fusion,* vol. 2, pp. WED5/19-WED5/25, July 2000.

[102] J.Z. Sasiadek, "Sensor Fusion," *Ann. Rev. in Control 26,* pp. 203-228, 2002.

[103] A.P. Dempster, "A Generalisation of Bayesian Inference," *J. Royal Statistical Soc.,* vol. 30, pp. 205-247, 1968.

[104] G. Shafer, *A Mathematical Theory of Evidence.* Princeton Univ. Press, 1976.

[105] J. Jiu, Y. Tan, and X. Yang, "A Framework for Identification via Multisensor Data Fusion," *Proc. 2003 Int'l Symp. Intelligent Control,* 2003.

[106] N.-H. Yu and Y. Yin, "Multiple Level Parallel Decision Fusion Model with Distributed Sensors Based on Dempster-Shafer Evidence Theory," *Proc. Second Int'l Conf. Machine Learning and Cybernetics,* Nov. 2003.

[107] T. Li and K.S. Ishwar, "Optimal Multiple Level Decision Fusion with Distributed Sensors," *IEEE Trans. Aerospace and Electronic Systems,* vol. 29, no. 4, Oct. 1993.

[108] N.R. Mudigonda, R. Kacalenga, and D. Erickson, "The Application of Dempster-Shafer Theory for Landmine Detection," *Proc. SPIE Multisensor, Multisource Information Fusion, Architectures, Algorithms, and Applications,* pp. 103-112, 2003.

[109] R.R. Murphy, "Dempster-Shafer Theory for Sensor Fusion in Autonomous Mobile Robots," *IEEE Trans. Robotics and Automation,* vol. 14, no. 2, pp. 197-206, Apr. 1998.

[110] B.R. Bracio, W. Horn, and D.P.F. Moller, "Sensor Fusion in Biomedical Systems," *Proc. IEEE 19th Ann. Int'l Conf., Eng. in Medicine and Biology Soc.* vol. 3, pp. 1387-1390, 1997.

[111] S. Challa and D. Koks, "Bayesian and Dempster-Shafer Fusion," *Sadhana,* vol. 29, part 2, pp. 145-174, Apr. 2004.

[112] S.P. Chaudhuri and S. Das, "Neural Networks for Data Fusion," *Proc. IEEE Int'l Conf. Systems Eng.*, 1990.
[113] S.A. Storm, "An Investigation into the Effects of Correlation in Sensor Fusion," thesis, Dept. Air Force, Air Univ., US Air Force Inst. of Technology, Mar. 2003.
[114] J. Kittler, "Multi-Sensor Integration and Decision Level Fusion," *Proc. DERA/IEE Workshop Intelligent Sensor Processing (Ref. No. 2001/050),* pp. 6/1-6/6, 2001.
[115] J. Kittler and F.M. Alkoot, "Sum Versus Vote Fusion in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, Jan. 2003.
[116] J. Kittler and K. Messer, "Fusion of Multiple Experts in Multimodal Biometric Personal Identity Verification Systems," *Proc. 2002 12th IEEE Workshop Neural Networks for Signal Processing,* 2002.
[117] R. Kacalenga, D. Erickson, and D. Palmer, "Voting Fusion for Landmine Detection," *IEEE Aerospace and Electronic Systems Magazine,* vol. 18, no. 8, pp. 13-19, Aug. 2003.
[118] L.A. Klein, "A Boolean Algebra Approach to Multiple Sensor Voting Fusion," *IEEE Trans. Aerospace and Electronic Systems,* vol. 29, no. 2, Apr. 1993.
[119] M.J. Caruso, "Applications of Magnetic Sensors for Low Cost Compass Systems," *Proc. IEEE 2000 Position, Location, and Navigation Symp.,* pp. 174-184, 2000.
[120] C.S. Raghavendra, C. Meesookho, and S. Narayanan, "Collaborative Classification Applications in Sensor Networks," *Proc. Second IEEE Sensor Array and Multichannel Signal Processing Workshop,* Aug. 2002.
[121] M. Duarte and Y.-H. Hu, "Vehicle Classification in Distributed Sensor Networks," *J. Parallel and Distributed Computing,* vol. 64, no. 7, pp. 826-838, 2004.
[122] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Line in the Sand—Wireless Sensor Network for Target Detection, Classification and Tracking," *Computer Networks,* vol. 46, pp. 605-634, 2004.
[123] R.R. Brooks, P. Ramarathan, and A.M. Sayeed, "Distributed Target Classification Tracking in Sensor Networks," *Proc. IEEE,* vol. 91, no. 8, Aug. 2003.
[124] T. Clouqueur, P. Ramanathan, K.K. Saluja, and K.-C. Wang, "Value-Fusion Versus Decision-Fusion for Fault-Tolerance in Collaborative Target Detection in Sensor Networks," *Proc. Fusion 2001 Conf.,* 2001.
[125] M.L. Hinman, "Some Computational Approaches for Situation Assessment and Impact Assessment," *Proc. Fusion 2002,* vol. 1, pp. 687-693, 2002.
[126] C.G. Looney and L.R. Liang, "Cognitive Situation and Threat Assessments of Ground Battlespaces," *Information Fusion,* vol. 4, pp. 297-308, 2003.
[127] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor Fusion Using Dempster-Shafer Theory," *Proc. IEEE Instrumentation and Measurement Technology Conf.,* pp. 21-23, May 2002.
[128] K. Wallenius, "Support for Situation Awareness in Command and Control," *Proc. Seventh Int'l Conf. Information Fusion,* pp. 1117-1124, 2004.
[129] G. Jakobson, L. Lewis, and J. Buford, "An Approach to Integrated Cognitive Fusion," *Proc. Seventh Int'l Conf. Information Fusion,* pp. 1210-1217, 2004.
[130] S. Sitharama Iyengar and Q. Wu, "Computational Aspects of Distributed Sensor Networks," *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks (ISPAN '02),* pp. 19-26, 2002.
[131] D.E. Penny, "Multisensor Management for Passive Target Tracking in an Anti Submarine Warfare Scenario," techinical report, United Kingdom Defence Evaluation and Research Agency, 1999.
[132] D.E. Penny, "Sensor Management in an ASW Data Fusion System, Sensor Fusion: Architectures, Algorithms and Applications III," *Proc. SPIE,* vol. 3719, 1999.
[133] M.L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "Multisensor Resource Management Using Posterior Cramer-Rao Lower Bounds," *IEEE Trans. Aerospace and Electronic Systems,* vol. 40, no. 2, Apr. 2004.
[134] R. Niu, P. Varshney, K. Mehrotra, and C. Mohan, "Temporal Fusion in Multi-Sensor Target Tracking Systems," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 2, pp. 1030-1037, 2002.
[135] R. Niu, P. Varshney, K. Mehrotra, and C. Mohan, "Sensor Staggering in Multi-Sensor Target Tracking Systems," *Proc. 2003 IEEE Radar Conf.,* 2003.
[136] N. Xiong and P. Svensson, "Multisensor Management for Information Fusion: Issues and Approaches," *Information Fusion,* vol. 3, pp. 163-186, 2002.
[137] K. Hintz and E. McVey, "A Measure of the Information Gain Attributable to Cuing," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 21, no. 2, pp. 434-442, Mar.-Apr. 1991,
[138] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management a Decentralized Information-Theoretic Approach.* Ellis Horwood, 1994.
[139] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-Based Sensor Selection Heuristic for Target Localisation," *Proc. Third Int'l Symp. Information Processing in Sensor Networks,* pp. 36-45, Apr. 2004,
[140] J. Moore, T. Keiser, R. Brooks, S. Phoha, D. Friedlander, J. Koch, A. Reggio, and N. Jacobson, "Tracking Targets with Self-Organising Distributed Ground Sensors," *Proc. 2003 Aerospace Conf.,* vol. 5, pp. 5-2113-5-2123, 2003.
[141] D. Friedlander, C. Griffin, N. Jacobson, S. Phoha, and R. Brooks, "Dynamic Agent Classification Using Ad Hoc Mobile Acoustic Sensor Network," *EURASIP J. Applied Signal Processing,* no. 4, pp. 371-377, 2003.
[142] L.-K. Soh, C. Tsatoulis, and H. Sevay, "A Satisficing, Negotiated and Learning Coalition Formation Architecture," *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz Jr., and M. Tambe, eds., pp. 109-138, Kluwer Academic, 2003.
[143] B. Horling, R. Mailer, J. Shen, R. Vincent, and V. Lesser, "Using Autonomy, Organizational Design and Negotiation in a Distributed Sensor Network," *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz Jr., and M. Tambe, eds., pp. 139-184, Kluwer Academic 2003.
[144] O. Yadgar, S. Kraus, and C.L. Ortiz Jr., "Scaling-Up Distributed Sensor Networks: Cooperative Large-Scale Mobile-Agent Organizations," *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz Jr., and M. Tambe, eds., pp. 185-218, Kluwer Academic, 2003.
[145] C.L. Ortiz Jr., T.W. Rauenbusch, E. Hsu, and R. Vincent, "Dynamic Resource-Bounded Negotiation in Non-Additive Domains," *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz, Jr., and M. Tambe, eds., pp. 61-107, Kluwer Academic, 2003.
[146] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine,* vol. 40, no. 8, pp. 102-114, Aug. 2002.
[147] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks 38,* pp. 393-422, 2002.
[148] C.A. Stelios, S.C.A. Thomopoulos, and L. Zhang, "Distributed Filtering with Random Sampling and Delay," *Proc. 27th IEEE Conf. Decision and Control,* vol. 3, pp. 2348-2353, Dec. 1988.
[149] H.L. Alexander, "State Estimation for Distributed Systems with Sensing Delay," *Proc. Int'l Soc. Optical Eng.,* vol. 1470, pp. 103-111, Aug. 1991.
[150] T.D. Larsen, N.A. Andersen, O. Ravn, and N.K. Poulson, "Incorporation of Time Telayed Measurements in a Discrete-Time Kalman Filter," *Proc. 37th IEEE Conf. Decision and Control,* vol. 4, pp. 3972-3977, 1998.
[151] S.S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Boston: Artech House, 1999.
[152] R.D. Hilton, D.A. Martin, and W.D. Blair, "Tracking with Time Delayed Multisensor Systems," Document Number NSWCDD/TR-9e/351, Naval Surface Warfare Center, Dahlgren, 1993.
[153] Y. Bar-Shalom, "Update with Out-of-Sequence Measurements in Tracking: Exact Solution," *IEEE Trans. Aerospace and Electronic Systems,* vol. 38, pp. 769-778, July 2002.
[154] M. Mallick, S. Coraluppi, and C. Carthel, "Advances in Asynchronous and Decentralized Estimation," *Proc. 2001 IEEE Aerospace Conf.,* Mar. 2001.
[155] E.W. Nettleton and H. Durrant-Whyte, "Delayed and Asequent Data in Decentralized Sensing Networks," *Proc. SPIE Conf.,* vol. 4571, Oct./Nov. 2001.
[156] K.S. Zhang, X. R. Li, and Y.M. Zhu, "Optimal Update with Out of Sequence Updates for Distributed Filtering," *Proc. Fifth Int'l Conf. Information Fusion,* July 2002.
[157] Y. Bar-Shalom, H. Chen, and M. Mallick, "One Step Solution for the Multistep Out-of-Sequence Measurement Problem in Tracking," *IEEE Trans. Aerospace and Electronic Systems,* vol. 40, no. 1, Jan. 2004.

[158] Y. Bar-Shalom, M. Mallick, H. Chen, and R. Washburn, "One-Step Solution for the General Out-of-Sequence Measurement Problem in Tracking," *Proc. 2002 IEEE Aerospace Conf.,*, vol. 4, pp. 1551-1559, 2002.
[159] L. Hong, S. Cong, and D. Wicker, "Distributed Multi-Rate Interacting Multiple Model (DRIMM) Filtering with Out-of-Sequence GMTI Data," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 2, pp. 1054-1061, 2002.
[160] L. Hong, S. Cong, and D. Wicker, "Distributed Multirate Interacting Multiple Model Fusion (DMRIMMF) with Application to Out-of-Sequence GMTI Data," *IEEE Trans. Automatic Control,* vol. 49, no. 1, Jan. 2004.
[161] L. Hong, S. Cong, and D. Wicker, "Multirate Interacting Multiple Model (MRIMM) Filtering with Out-of-Sequence GMTI Data," *IEE Proc. Radar, Sonar, and Navigation,* vol. 150, pp. 333-343, 2003.
[162] X. Wang and S. Challa, "Augmented State IMM-PDA for OOSM Solution to Maneuvering Target Tracking in Clutter," *Proc. Int'l Radar Conf.,* pp. 479-485, 2003.
[163] M. Mallick, S.J. Krant, and Y. Bar-Shalom, "Multi-Sensor Multi-Target Tracking Using Out-of-Sequence Measurements," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 1, pp. 135-142, 2002.
[164] M. Orton and A. Marrs, "A Bayesian Approach to Multi-Target Tracking and Data Fusion with Out of Sequence Measurements," *Proc. IEE Int'l Seminar Target Tracking: Algorithms and Applications,* pp. v1-15, 2001.
[165] M.L. Hernandez, A.D. Marrs, S. Maskell, and M.R. Orton, "Tracking and Fusion for Wireless Sensor Networks," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 2, pp. 1023-1029, 2002.
[166] M. Mallick, T. Kirubarajan, and S. Arulampalam, "Out-of-Sequence Processing Using Particle Filters," *Proc. IEEE Aerospace Conf.,* vol. 4, pp. 1809-1818, 2002.
[167] M. Mallick and A. Marrs, "Comparison on of the KF and Particle Filter Based Out-of-Sequence Measurement Filtering Algorithms," *Proc. Sixth Int'l Conf. Information Fusion,* vol. 1, pp. 422-429, 2003.
[168] S. Challa, X. Wang, and J. Legg, "Track-to-Track Fusion Using Out-of-Sequence Tracks," *Proc. Int'l Conf. Information Fusion,* pp. 919-926, 2002.
[169] S. Challa, J.E. Robin, and X. Wang, "A Bayesian Solution and Its Approximations to Out-of-Sequence Measurement Problems," *Information Fusion,* vol. 4, no. 3, pp. 185-199, 2003.
[170] P. Maybeck, *Stochastic Models, Estimation and Control.* Academic Press, 1979.
[171] S.-L. Sun, "Multi-Sensor Optimal Information Fusion Kalman Filter for Discrete Multichannel ARMA Signals," *Proc. 2003 IEEE Int'l Symp. Intelligent Control,* pp. 377-382, 2003.
[172] S.-L. Sun and Z.-L. Deng, "Multi-Sensor Optimal Information Fusion Kalman Filter," *Automatica,* vol. 40, pp. 1017-1023, 2004.
[173] S.-L. Sun, "Multi-Sensor Optimal Information Fusion Kalman Filters with Applications," *Aerospace Science and Technology,* vol. 8, pp. 57-62, 2004.
[174] Y. Bar-Shalom, "On the Track to Track Correlation Problem," *IEEE Trans. Automatic Control,* vol. 26, no. 2, Apr. 1981.
[175] B.F. La Scala and A. Farina, "Choosing a Track Association Method," *Information Fusion,* vol. 3, pp. 119-133, 2001.
[176] J.K. Uhlmann, S. Julier, and H.F. Durrant-Whyte, "A Culminating Theory in the Theory and Practice of Data Fusion, Filtering and Decentralized Estimation," technical report, Covariance Intersection Working Group, 1997.
[177] B. La Scala and A. Farina, "Effects of Cross-Covariance and Resolution on Track Association," *Proc. Third Int'l Conf. Information Fusion,* vol. 2, pp. WED1/10-WED1/16, 2000.
[178] A.D. Marrs, C.M. Reed, A.R. Webb, and H.C. Webber, "Data Incest and Symbolic Information Processing," technical report, Inited Kingdom Defence Evaluation and Research Agency, Mar. 1999.
[179] D. Nicholson, C. Lloyd, S. Julier, and J. Uhlmann, "Scalable Distributed Data Fusion," *Proc. Fifth Int'l Conf. Information Fusion,* vol. 1, pp. 630-635, 2002.
[180] L.Y. Pao, "Distributed Multisensor Fusion," Am. Inst. of Aeronautics and Astronautics, 1994.
[181] L.Y. Pao and M. Kalandros, "Algorithms for Distributed Architecture Tracking," *Proc. Am. Control Conf.,* June 1997.
[182] W. Khawsuk and L.Y. Pao, "Decorrelated State Estimation for Distributed Tracking of Interacting Targets in Cluttered Environments," *Proc. Am. Control Conf.,* May 2002.
[183] W. Khawsuk and L.Y. Pao, "Decorrelated State Estimation for Distributed Tracking Using Multiple Sensors in Cluttered Environments," *Proc. 2003 Am. Control Conf.,* vol. 4, pp. 3208-3214, 2003.
[184] L. Trailovic and L.Y. Pao, "Variance Estimation and Ranking of Gaussian Mixture Distributions in Target Tracking Applications," *Proc. 41st IEEE Conf. Decision and Control,* vol. 2, pp. 2195-2201, 2002.
[185] S.P. McLaughlin, R.J. Evans, and V. Krishnamurthy, "Data Incest Removal in a Survivable Estimation Fusion Architecture," *Proc. Sixth Int'l Conf. Information Fusion,* vol. 1, pp. 229-236, 2003.
[186] S.P. McLaughlin, V. Krishnamurthy, and S. Challa, "Managing Data Incest in a Distributed Sensor Network," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 5, pp. 269-272, 2003.
[187] S.J. Julier and J.K. Uhlmann, "A Non-Divergent Algorithm in the Presence of Unknown Correlation," *Proc. Am. Control Conf.,* 1997.
[188] S. Grime and H.F. Durrant-Whyte, "Data Fusion in Decentralized Sensor Networks," *Control Eng. Practice,* vol. 2, no. 5, pp. 849-863, Oct. 1994.
[189] P.O. Arambel, C. Rago, and R.K Mehra, "Covariance Intersection Algorithm for Distributed Spacecraft State Estimation," *Proc. 2001 Am. Control Conf.,* vol. 6, pp. 4398-4403, 2001.
[190] M.B. Hurley, "An Information Theoretic Justification for Covariance Intersection and Its Generalization," *Proc. Fifth Int'l Conf. Information Fusion,* pp. 505-511 2002.
[191] L. Chen, P.O. Arambel, and R.K. Mehra, "Fusion under Unknown Correlation—Covariance Intersection as a Special Case," *Proc. Fifth Int'l Conf. Information Fusion,* pp. 905-912, 2002.
[192] L. Chen, P.O. Arambel, and R.K. Mehra, "Fusion Under Unknown Correlation: Covariance Intersection Revisited," *IEEE Trans. Automatic Control,* vol. 47, no. 11, Nov. 2002.
[193] W. Niehsen and R. Bosch, "Information Fusion Based on Fast Covariance Intersection Filtering," *Proc. Fifth Int'l Conf. Information Fusion,* pp. 901-904, 2002.
[194] J.K. Uhlmann, "Covariance Consistency Methods for Fault-Tolerant Distributed Data Fusion," *Information Fusion* vol. 4, pp. 201-215, 2003.
[195] A.R. Benaskeur, "Consistent Fusion of Correlated Data Sources," *Proc. IEEE 2002 28th Ann. Conf. Industrial Electronics Soc.,* vol. 4, pp. 2652-2656, Nov. 2002.

Duncan Smith received the degree in computer science from the University of Leeds in 2001. After graduating, he joined QinetiQ, where he has worked ever since as a research scientist, developing fusion algorithms for the Royal Navy. Since 2004, he has also been studying for the PhD degree at Loughborough University. His research interests include target tracking, data fusion, neural networks, evolutionary algorithms, and real ale.

Sameer Singh (M'98) received the BE degree with distinction in computer engineering from the Birla Institute of Technology, Birla, India, the MS degree in information technology for manufacturing from the University of Warwick, Warwick, United Kingdom, and the PhD degree in speech and language analysis of stroke patients from the University of the West of England, Bristol, United Kingdom. His main research interests are in image processing, medical imaging, computer vision, and pattern recognition. He is the director of the Research School of Informatics at Loughborough University, United Kingdom. He is also the chairman of the British Computer Society Specialist group on Pattern Analysis and Robotics. In his editorial capacity, he serves as the editor-in-chief of the *Pattern Analysis and Applications Journal* and as an associate editor of the following journals: *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Real-Time Image Analysis, Pattern Recognition, Knowledge and Information Systems,* and *Neural Computing and Applications*. He is on the executive advisory board of the *IEEE Transactions on Fuzzy Computing*. He is a fellow of the Royal Statistical Society, and a member of the IEEE, BMVA-IAPR, and IEE (CEng).