Efficient Multi-Source Data Fusion for Decentralized Sensor Networks

Philip J. Haney[*], Christopher M. Lloyd[†]

[*]BAE Systems – Advanced Systems & Technology
130 Daniel Webster Highway, Merrimack, NH 03054

[†]BAE Systems - Advanced Technology Centre
Sowerby Building, Bristol BS34 7QW, UK

philip.j.haney@baesystems.com chris.m.lloyd@baesystems.com

ABSTRACT

A highly-scalable, Bayesian approach to the problem of performing multi-source data fusion and target tracking in decentralized sensor networks is presented. Previous applications of decentralized data fusion have generally been restricted to uni-modal/uni-source sensor networks using Gaussian based approaches, such as the Kalman or information filter. However, with recent interest to employ complex, multi-modal/multi-source sensors which potentially exhibit observation and/or process non-linearities along with non-Gaussian distributions, the need to develop a more generalized and scalable method of decentralized data fusion is required. The probabilistic approach featured in this work provides the ability to seamlessly integrate and efficiently fuse multi-source sensor data in the absence of any linearity and/or normality constraints. The proposed architecture is fully decentralized and provides a methodology that scales extremely well to any growth in the number of targets or region of coverage. This paper will illustrate that our multi-source data fusion architecture is capable of providing high-precision tracking performance in complex, non-linear/non-Gaussian operating environments. In addition, we will also show that our architecture provides an unprecedented scaling capability for decentralized sensor networks as compared to similar architectures which communicate information using particle data, Gaussian mixture models or Parzen density estimators.

1.0 INTRODUCTION

In the field of multi-sensor data fusion, decentralized data fusion has become an attractive alternative to centralized data fusion primarily due to the inherent robustness and scalability features that decentralized architectures offer. In its most primitive form, a decentralized sensor network involves processing capability at each sensor - eliminating the need and subsequent vulnerability of a central processing node - along with the capacity for each sensor to efficiently communicate its information to neighboring sensors without requiring any knowledge of the network topology [1].

To date, the majority of fielded implementations utilizing decentralized data fusion have relied on linear/Gaussian assumptions and the Kalman/information filter [2,3,4,5,6]. Even though such systems have produced impressive results, the natural desire to utilize a wider mixture of more complex sensor types - potentially exhibiting observation and/or process non-linearities along with non-Gaussian distributions - has

Haney, P.J.; Lloyd, C.M. (2006) Efficient Multi-Source Data Fusion for Decentralized Sensor Networks. In *Battlefield Acoustic Sensing for ISR Applications* (pp. 13-1 – 13-18). Meeting Proceedings RTO-MP-SET-107, Paper 13. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

generated a need for more generalized information fusion techniques. A variety of methods have been applied to the problem of non-linear/non-Gaussian decentralized data fusion in which the majority of such methods have been focused on particle filters [7,8], Gaussian mixture models or Parzen density estimators [9,10] or some combination of the two [11,12,13]. All of the aforementioned techniques are capable of performing generalized estimation, however, not all techniques lend themselves well to decentralized applications as a result of the scalability concerns they inherently generate.

This paper presents a highly-scalable, Bayesian approach to the problem of performing generalized, multi-source data fusion and target tracking in decentralized sensor networks. A brief background of linear decentralized data fusion is first provided in Section 2, and performance results related to currently fielded linear decentralized data fusion implementations are provided in Section 3[1]. Section 4 introduces a proposed approach to performing probabilistic decentralized data fusion, and corresponding simulated performance results are presented in Section 5[2]. Section 6 summarizes the paper and provides direction for future efforts.

2.0 LINEAR DECENTRALIZED DATA FUSION

2.1 Data fusion in sensor networks

When formulated from a Bayesian standpoint, the data fusion problem is to recursively compute the posterior probability density function (PDF) of the state of interest x_k at time step k, using the measurement set $Z^k = \{z_j, j=1,2,...,k\}$; this can be calculated using Bayes rule

$$p(x_k|Z^k) = \frac{p(z_k|x_k)p(x_k|Z^{k-1})}{p(z_k|Z^{k-1})} \tag{2.1.1}$$

where $p(z_k|x_k)$ is the measurement likelihood and $p(x_k|Z^{k-1})$ is the predicted PDF at time k computed according to (2.1.2). The denominator of (2.1.1) acts as a normalisation factor.

$$p(x_k|Z^{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z^{k-1})dx_{k-1} \tag{2.1.2}$$

When the underlying target distribution and likelihood are assumed to be Gaussian, equations (2.1.1), (2.1.2) reduce to the Kalman filter [14] and its Information filter equivalent.

To allow the global state estimate to be computed in a decentralised system – where only communication with nearest neighbours is allowed – nodes must communicate their own data *and* data received from other sensors. Care must be taken to prevent nodes receiving data along multiple paths. If not correctly handled the re-fusion of information in this way gives rise to the rumour propagation problem which can seriously degrade the quality of fusion [15]. It is also desirable that each node should not have to record the provenance of every piece of information generated by any sensor. For tree connected communication networks the solution to this problem is the Channel filter [16]. The Channel filter is a probability distribution conditioned on the common information held between a pair of nodes. When Channel filters are used to identify redundant information, only posterior distributions need to be transmitted between nodes. This allows the computation of the global posterior without sending measurements or provenance information.

[1] Sections 2 and 3 are provided by BAE Systems - Advanced Technology Centre.
[2] Sections 4 and 5 are provided by BAE Systems - Advanced Systems & Technology.

If nodes *a* and *b* communicate their information, the updated estimate can be calculated as the product of their distributions divided by the common information [12].

$$p(x_k \mid Z_k^a \cup Z_k^b) = \frac{p(x_k \mid Z_k^a)p(x_k \mid Z_k^b)}{p(x_k \mid Z_k^a \cap Z_k^b)} \tag{2.1.3}$$

When the underlying distributions are Gaussian, the channel filter is also an Information filter. Importantly, since information is additive, redundant information can be removed by subtraction. A discrete time model of a linear system $x_k = [x_1, x_2, \ldots, x_n]$, $x_i \in \Re$, is given by (2.1.4).

$$x_k = F_k x_{k-1} + G_k w_k \tag{2.1.4}$$

$$z_k = H_k x_k + v_k \tag{2.1.5}$$

where F_k is the state transition model; G_k projects the noise vector into the state, and w_k is a zero-mean, white noise vector with covariance Q_k. An observation of the state is made according to (2.1.5) where v_k is a zero mean white noise vector with covariance R_k.

For such a system the Kalman filter produces an optimal estimate $\hat{x}(k \mid k)$ of the true state $x(k)$ and an associated covariance $P(k \mid k) = \mathrm{E}[(\hat{x}(k \mid k) - x(k))'(\hat{x}(k \mid k) - x(k))]$ using the measurement set Z^k. The Kalman filter equations can be written in information form by introducing the identities (2.1.6) for the information vector and information matrix respectively.

$$\begin{aligned} y(k|k) &= P(k|k)^{-1}\hat{x}(k|k) \\ Y(k|k) &= P(k|k)^{-1} \end{aligned} \tag{2.1.6}$$

The Kalman filter update in information space is a simple addition of the predicted information matrices and vectors and the measurement written in information form (2.1.7), (2.1.8). The predicted densities can be calculated by converting the predicted state space (2.1.9) estimate to information space.

$$\begin{aligned} y(k|k) &= y(k|k-1) + i(k) \\ Y(k|k) &= Y(k|k-1) + I(k) \end{aligned} \tag{2.1.7}$$

$$\begin{aligned} i(k) &= H_k' R_k^{-1} z_k \\ I(k) &= H_k' R_k^{-1} H_k \end{aligned} \tag{2.1.8}$$

$$\begin{aligned} \hat{x}(k \mid k-1) &= F_k \hat{x}(k\text{-}1 \mid k\text{-}1) \\ P(k \mid k-1) &= F_k P(k-1 \mid k-1) F_k' + G_k Q_k G_k' \end{aligned} \tag{2.1.9}$$

The track-to-track update can now be written as the sum of information vectors and matrices, minus the channel filter (2.1.10).

$$\begin{aligned} y(k|k)^{+} &= y_a(k|k) + y_b(k|k) - y_{ab}(k|k-1) \\ Y(k|k)^{+} &= Y_a(k|k) + Y_b(k|k) - Y_{ab}(k|k-1) \end{aligned} \tag{2.1.10}$$

The channel filter is then updated to account for the new common information (2.1.11).

$$\begin{aligned} y_{ab}(k|k) &= y(k|k)^{+} \\ Y_{ab}(k|k) &= Y(k|k)^{+} \end{aligned} \tag{2.1.11}$$

Unfortunately, the channel filter can only be used in single connected networks. For arbitrary networks the Covariance Intersection (CI) algorithm can be used [17]. The CI algorithm is a convex combination of the information vectors and matrices.

$$\begin{aligned} y(k|k)^{+} &= \omega y_a(k|k) + (1-\omega) y_b(k|k) \\ Y(k|k)^{+} &= \omega Y_a(k|k) + (1-\omega) Y_b(k|k) \end{aligned} \tag{2.1.12}$$

where $\omega \in [0,1]$ is a free parameter to be optimised, typically to maximise the determinant of the resultant information matrix. The CI algorithm gives an estimate that is guaranteed to be consistent in the sense that the covariance is greater than the expected error. CI thus gains additional flexibility at the expense of optimality.

3.0 LINEAR DDF – FIELDED PERFORMANCE RESULTS

3.1 DDF Software Development Kit

The operations of a Decentralised Data Fusion (DDF) node, those of *predict*, *update*, *communicate* and *assimilate,* are independent of the underlying representation of the probability distribution. To facilitate the development and demonstration of novel DDF techniques, BAE Systems ATC has exploited this independence and developed a DDF Software Development Kit (SDK). Illustrated in Figure 1, the DDF SDK provides an overall DDF architecture with configuration, sensor and communication layer interfaces and support functions in an object-orientated framework written in C++. Initially the SDK utilised Kalman/Information filters, however the software has been written in a sufficiently general manner that any suitable representation such as particles [12] or mixtures of Gaussians [10] can be added.

3.2 Closed loop control

The DDF Software Development Kit (SDK) has been used by BAE Systems ATC to develop a real time closed loop sensor-to-actor control demonstration. The system integrates ATC CCTV cameras (Figure 2a), Robovolc mobile robotic capability (Figure 2b) and Unattended Ground Sensors (UGS) (Figure 2c). The scenarios considered were used to demonstrate the advantages of DDF within a heterogeneous sensor network in an end-to-end system that encompassed all sections of the OODA loop (Observe, Orient, Decide, and Act). The actuator was the Robovolc robot, the decision was which target(s) to intercept and when, and the orient stage involved both the physical orientation of the robot, and the sensor-to-target allocation for steer-able sensors.

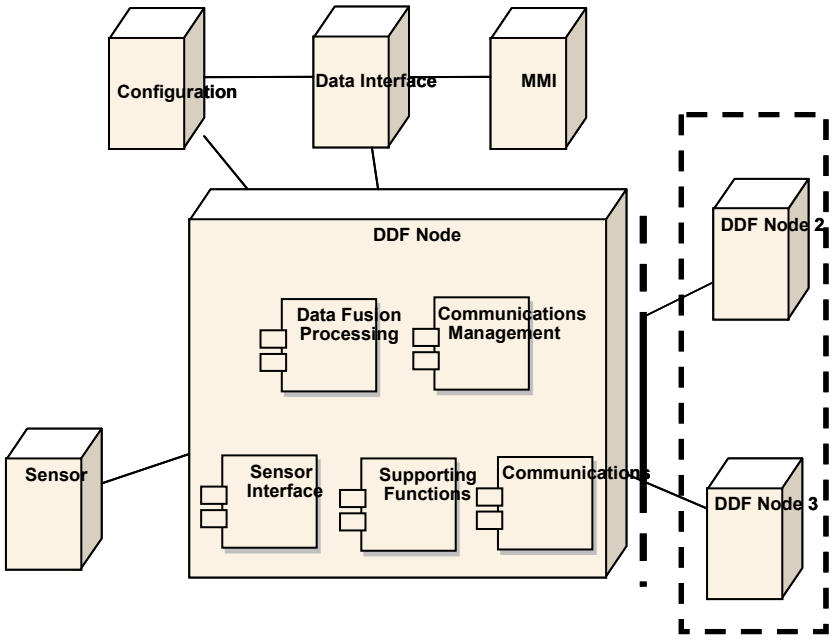
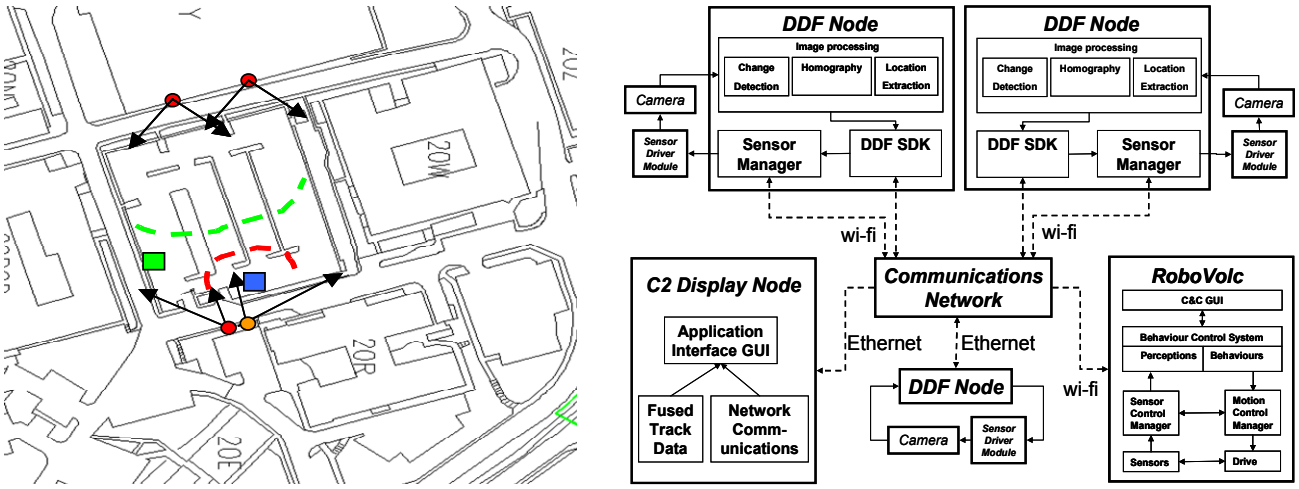


Figure 1: DDF SDK is a data-fusion software architecture

a b c

Figure 2: CCTV Camera (a), Robovolc (b), UGS (c)

The scenarios were representative of a number of military scenarios including UAV reconnaissance/ surveillance, anti-submarine warfare, urban asymmetrical warfare and mine counter measures. Figure 3a shows the experimental set-up. In the scenarios, pedestrians in the 9K car park give rise to tracks in the Common Operating Picture (COP). Robovolc, accessing a single DDF node associated with a CCTV camera (marked in orange in Figure 3a), defends a 'sensitive' region of the car park against intrusion (e.g. red area in Figure 3a). Robovolc defends a region from intrusion by physically interposing itself in the path of targets that appear to threaten the region. Once a pedestrian has been intercepted, Robovolc then returns to guarding the region. Robovolc moves to interception points supplied to it by software that sits between DDF and Robovolc (Figure 3b). This software analyses the DDF tracks and instructs Robovolc accordingly. The instruction to move is triggered by a target crossing an outer boundary shown in green in Figure 3a.



a b

Figure 3: Operational Setup (a), Software Architecture (b)

The UGS sensor (Figure 2c) was able to detect targets using its ultrasound tripwire while running the SDK DDF software at an acceptable rate indicating its ability to participate in a network as a full DDF sensor node.

In addition to the DDF tracking, sensor management was implemented across the network. Multi-sensor management using Decentralised Decision Making (DDM) algorithms was implemented across two steer-able sensors to provide a globally optimal sensor to target allocation. The manageable sensor was mimicked in software by only processing a subset of the cameras' Field Of View (FOV). This resulted in a narrower but steered FOV. The sensor management software controls the pointing of these software-steered cameras with a view to optimising the ability of Robovolc to defend the area. The results are drawn from analysis of the data collected during two experimentation periods, one for each implementation. The measures of effectiveness are drawn from the SIAP measures [18]. The relevant SIAP measure is completeness, and the relevant system metric is the number of successful (timely) intercepts. Figure 4 illustrates two still images from the demonstration video. In the upper right and left corners, the steerable camera FOVs are shown as light grey squares. The lower left corner illustrates a plan view showing the tracked target means and covariances.

Figure 4: The UGS was used as a trip wire to cue the CCTV FOV; the latter are the light grey squares

4.0 PROBABILISTIC DECENTRALIZED DATA FUSION

When applying linear decentralized data fusion to the problem of target tracking, a common observability/fusion platform in position space is typically required by all sensors in the network – whether it be in a spherical or Cartesian coordinate system. However, when multi-source sensors operating in disparate observability/fusion spaces are used to populate a network, a common platform for fusion no longer exists, and the linear decentralized data fusion algorithms must be abandoned for a more generalized estimation approach.

A generalized, probabilistic approach to performing multi-source data fusion and target tracking in decentralized sensor networks is illustrated in Figure 5. The proposed architecture provides the ability to seamlessly integrate and efficiently fuse multi-source sensor data in the absence of any linearity and/or normality constraints. In addition, the architecture is fully decentralized and provides a methodology that scales extremely well to any growth in the number of targets or region of coverage. The primary components of the described architecture are discussed in detail in the subsequent sections.

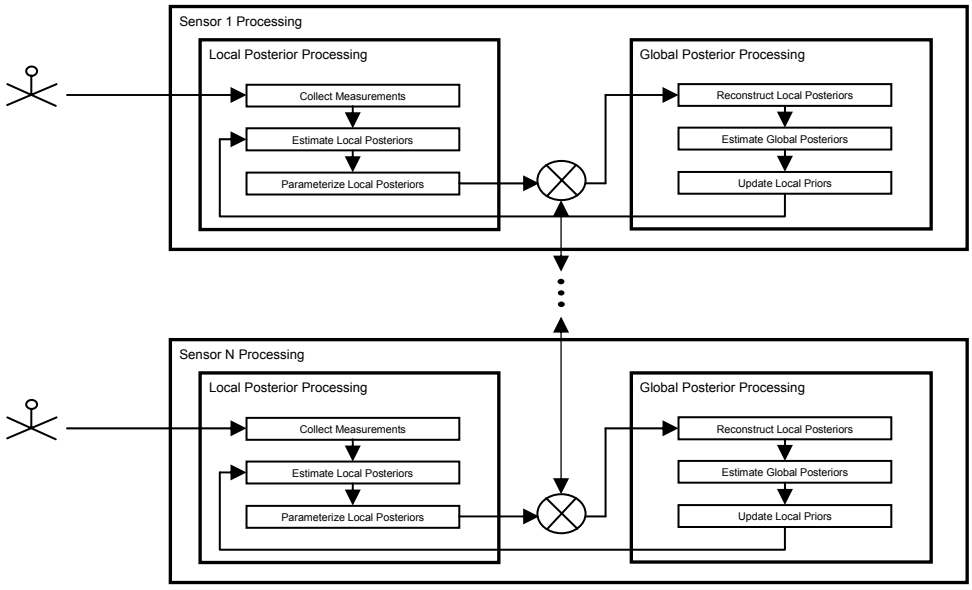


Figure 5: Probabilistic Decentralized Data Fusion Architecture

4.1 Particle Filters

The decentralized probabilistic approach outlined in Figure 5 utilizes particle filters to estimate the local posterior densities for each sensor. As described in the seminal paper by Gordon *et al.* [19] along with others [14,20], particle filters are based upon point mass representations of probability densities which can be applied to any state-space model in the absence of linearity and/or normality constraints. In the same way that a Kalman filter provides the optimal Bayesian estimate for the highly-restricted linear, Gaussian environment, a particle filter can be thought of as approximating the optimal Bayesian recursion without having to impose any linear model or noise distribution constraints.

The basic idea behind the particle filter is that the posterior density of interest is represented by a set of weighted particles, each of which forms an independent hypothesis for the state at a given time. If the weights are chosen correctly, this weighted set of particles becomes representative of the true posterior density in that expectations of the true posterior can be made arbitrarily close to the equivalent expectations of the set of weighted particles.

In order to describe its operation, assume that a particle set of size N is being used to approximate the posterior density of a state vector, x_k, based upon the set of all available measurements $Z^k=\{z_j, j=1,2,...,k\}$. At iteration $k-1$, the posterior density, $p(x_{k-1}|Z^{k-1})$, is represented by a set of particle values $(x^1_{k-1},x^2_{k-1},...,x^N_{k-1})$ and associated weights $(w^1_{k-1},w^2_{k-1},...,w^N_{k-1})$. At the next iteration, a new set of particle values $(x^1_k,x^2_k,...,x^N_k)$ and associated weights $(w^1_k,w^2_k,...,w^N_k)$ which characterize the posterior density, $p(x_k|Z^k)$, can be determined as follows:

$$x^i_k \sim p(x_k|x^i_{k-1}) \tag{4.1.1}$$

$$w^i_k = \frac{p(z_k|x^i_k)w^i_{k-1}}{\sum_{i=1}^{N} w^i_{k-1}p(z_k|x^i_k)} \tag{4.1.2}$$

where $p(x_k|x^i_{k-1})$ is the transition density and $p(z_k|x^i_k)$ is the likelihood function. Comparing (4.1.2) with (2.1.1) it can be seen that for a given particle,

$$w^i_{k-1} = p(x^i_k|Z^{k-1}) \tag{4.1.3}$$

$$w^i_k = p(x^i_k|Z^k). \tag{4.1.4}$$

As a result, from (2.1.1), (4.1.3) and (4.1.4), it is observed that the weights of a particle filter represent the prior and posterior densities corresponding to each particle's estimate of the state vector. Consequently, an overall estimate for the true posterior density can be theoretically obtained from the particle ensemble.

4.2 Estimating Local Posteriors

In this work, range-only, bearing-only and range/bearing sensors are evaluated for multi-source, decentralized data fusion in which particle filters are used to estimate the local posteriors in range/bearing space for all sensor types. The measurement vector, measurement covariance and particle filter state vector used for all sensor types are defined as

$$\underline{z}_k = [z_{r,k}, z_{\theta,k}]', \quad R_k = diag(\sigma^2_{z_{r,k}}, \sigma^2_{z_{\theta,k}}), \quad \underline{x}_k = [r_k, \theta_k, \dot{r}_k, \dot{\theta}_k]'. \tag{4.2.1}$$

Using (4.2.1), a generalized likelihood function can be defined as the product of the individual range/bearing likelihoods for a given sensor type as follows:

1. Range-Only Sensor - $p(\underline{z}_k|\underline{x}_k) = p(z_{r,k}|\underline{x}_k)p(z_{\theta,k}|\underline{x}_k) = N(z_{r,k}, \sigma^2_{z_{r,k}})U(\theta_{min},\theta_{max})$
2. Bearing-Only Sensor - $p(\underline{z}_k|\underline{x}_k) = p(z_{r,k}|\underline{x}_k)p(z_{\theta,k}|\underline{x}_k) = U(r_{min},r_{max})N(z_{\theta,k}, \sigma^2_{z_{\theta,k}})$ (4.2.2)

3. Range/Bearing Sensor - $p(\underline{z}_k \mid \underline{x}_k) = p(z_{r,k} \mid \underline{x}_k)p(z_{\theta,k} \mid \underline{x}_k) = N(z_{r,k}, \sigma^2_{z_{r,k}})N(z_{\theta,k}, \sigma^2_{z_{\theta,k}})$

where $N(z, \sigma^2)$ corresponds to a normal distribution with mean z and variance σ^2, and $U(\alpha_{min}, \alpha_{max})$ corresponds to a uniform distribution over the region α_{min} to α_{max}.

The generalized transition density used for this work is based on a zero-mean, first-order Markov model [21] and is defined by the following state transition matrix and process noise covariance

$$\Phi = \begin{bmatrix} 1 & 0 & \tau_r(1-e^{-\Delta t/\tau_r}) & 0 \\ 0 & 1 & 0 & \tau_\theta(1-e^{-\Delta t/\tau_\theta}) \\ 0 & 0 & e^{-\Delta t/\tau_r} & 0 \\ 0 & 0 & 0 & e^{-\Delta t/\tau_\theta} \end{bmatrix},$$

$$Q_k = \begin{bmatrix} 2\sigma_r^2/\tau_r & 0 & 0 & 0 \\ 0 & 2\sigma_\theta^2/\tau_\theta & 0 & 0 \\ 0 & 0 & 2\sigma_r^2/\tau_r & 0 \\ 0 & 0 & 0 & 2\sigma_\theta^2/\tau_\theta \end{bmatrix} \begin{bmatrix} \Delta t^3/3 & 0 & \Delta t^2/2 & 0 \\ 0 & \Delta t^3/3 & 0 & \Delta t^2/2 \\ \Delta t^2/2 & 0 & \Delta t & 0 \\ 0 & \Delta t^2/2 & 0 & \Delta t \end{bmatrix} \quad (4.2.3)$$

where τ_r, τ_θ, σ_r, and σ_θ are tunable parameters related to the expected target motion in range/bearing space - each of which can assume different values depending upon the amount and type of fusion which has occurred for a specific posterior.

Applying the generalized likelihood function and transition density defined in (4.2.2) and (4.2.3) to (4.1.1) and (4.1.2), the particle filter is observed to provide a consistent method of local posterior estimation for the multi-source sensor types evaluated in this work.

4.3 Parameterizing Local Posteriors

Communication of information is paramount for any form of distributed data fusion. Consequently, a fundamental requirement for decentralized fusion is efficient scaling of communicated information relative to any growth in the number of targets or region of coverage. To date, the majority of methods investigated for communicating information in decentralized architectures have been focused on particle set optimization [7,8], Gaussian mixture models or Parzen density estimators [9,10] or some combination of the two [11,12,13]. All of these methods, however, are prone to scalability concerns as the number of particles, GMM components or Parzen components required to adequately estimate and/or reconstruct posteriors of interest necessitate considerable increase as the number of targets or region of coverage escalate. Our solution to this scalability concern is rooted in high-order statistics and the generalized lambda distribution (GLD) - a four-parameter probability distribution whose density function can assume a wide variety of uni-modal, symmetric and/or asymmetric shapes [22,23][3].

[3] The uni-modal limitation of the generalized lambda distribution prevents true, decentralized multi-modal estimation. However, for applications such as target tracking where a uni-modal assumption for the marginal distributions of interest may not be too erroneous, the generalized lambda distribution provides an intermediate solution to the decentralized scalability concern until a comparable multi-modal version can be developed.

Derived from Tukey's original lambda distribution defined by the quantile function

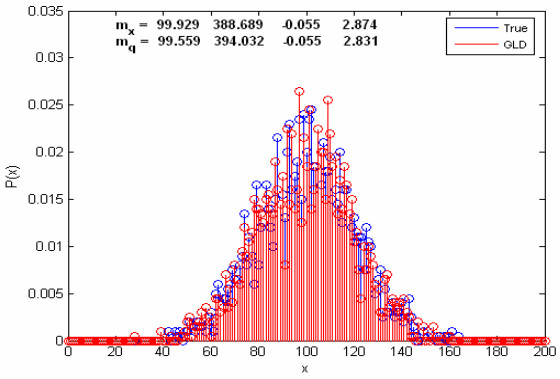
$$Q(u) = \begin{cases} u^{\lambda} - (1-u)^{\lambda} & ,\lambda \neq 0 \\ \dfrac{log(u)}{(1-u)} & ,\lambda = 0 \end{cases} \tag{4.3.1}$$

for $0 \leq u \leq 1$ [24], the generalized lambda distribution parameterizes (4.3.1) as follows

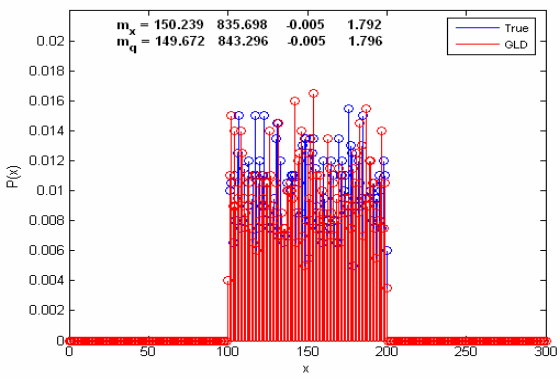
$$Q(u) = \lambda_1 + \frac{u^{\lambda_3} - (1-u)^{\lambda_4}}{\lambda_2} \tag{4.3.2}$$

where λ_1 acts as a location parameter, λ_2 acts as a scale parameter, and the combination of λ_3 and λ_4 jointly capture the shape of the empirical distribution.

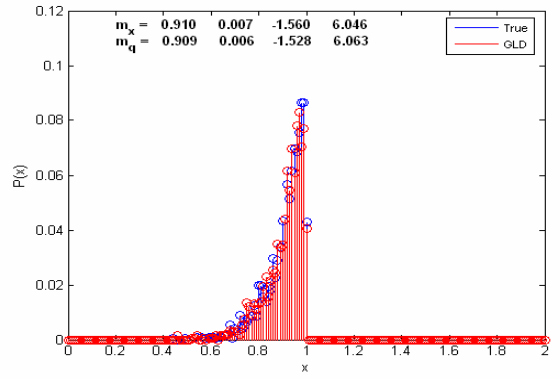
In order to describe a probability distribution using the generalized lambda distribution, the λ-parameters are typically matched to the first four moments of a desired data distribution using a methodology similar to the one outlined in [22]. Figure 6 illustrates the ability of the generalized lambda distribution to accurately estimate various uni-modal data distributions. The first four moments of the original data distribution and those of the GLD distribution are annotated on each plot. From Figure 6, the mean percent moment error is observed to be less than 1% for all GLD estimated distributions.



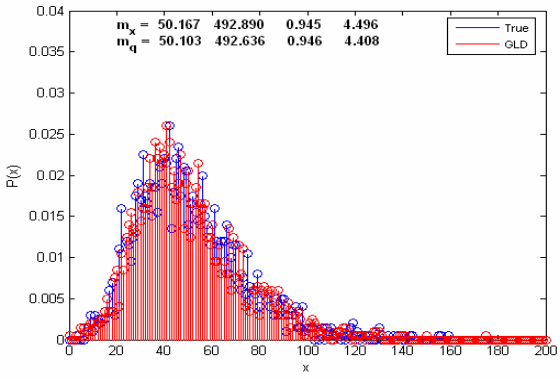
Normal (100,20)



Uniform (100,200)



Beta (10,1)



Gamma (5,10)

Figure 6: GLD Estimation of Various Uni-Modal Distributions

Table 1 – taken from [11,12] and slightly modified – illustrates the bandwidth requirements for communicating:

- a particle set
- a GMM approximation of a particle set
- a Parzen density approximation of a particle set
- a GLD approximation of a particle set

In [11,12] it was stated that 20 GMM components and 50 Parzen components were required to adequately approximate a set of 2000 particles. Extending these parameters as hard limits for accurate posterior representation and noting that the GLD requires 4 components per dimension of the state vector, Table 1 illustrates the number of posteriors that can be accurately estimated for a given state vector dimensionality and a given amount of available bandwidth using the following formula:

$$\text{Number of Posteriors Accurately Communicated} = \frac{\text{Available Bandwidth (floats)}}{\left(\frac{\text{Required Bandwidth (floats)}}{\text{Component}}\right)\left(\frac{\text{Required Components}}{\text{Accurate Posterior Representation}}\right)}.$$

From Table 1 it can be seen that the generalized lambda distribution provides a significantly more compact method of posterior representation as compared to all the other methods considered.

Table 1: Bandwidth Requirements

Representation	Dimension	Available Bandwidth (floats)	Required Bandwidth (floats) per Component	Required Components per Accurate Posterior Representation	Number of Posteriors Accurately Communicated
Particle	4	500	1	2000	0
GMM	4	500	14	20	1
Parzen	4	500	5	50	2
GLD	4	500	1	16	31
Particle	6	500	1	2000	0
GMM	6	500	27	20	0
Parzen	6	500	7	50	1
GLD	6	500	1	24	20

4.4 Reconstructing Local Posteriors

One advantage of using a quantile function like (4.3.2) to describe a distribution lies in its ability to efficiently generate random variates. For instance, if Q is the quantile function for a specific probability distribution, then through the use of the inverse transformation method,

$$X = Q(u), \quad u \in (0,1), \tag{4.4.1}$$

random variates, X, with quantile function, Q, can be generated with little effort. Consequently, since sources of uniform random variates over the interval (0,1) are readily available, (4.4.1) provides an efficient method for generating random variates from distributions whose quantile functions are known and computationally tractable. As a result, and as demonstrated in Figure 6, this property coupled with (4.3.2) allows for efficient and accurate reconstruction of local posteriors communicated throughout a sensor network via a set of GLD parameters.

4.5 Estimating Global Posteriors

After the local posteriors have been parameterized, communicated and reconstructed at any receiving sensor(s), the number of targets and corresponding positions are estimated using an iterative, probabilistic approach. All posteriors at a given sensor are then associated with the estimated target information, and corresponding global posteriors are determined by fusing the associated local posteriors via multiplication of their joint probability densities.

Joint probability densities are typically estimated using histogram methods, however, such methods often introduce undesirable "artifact patterns" which hamper the global fusion process by introducing local extrema and obscuring the true global optimum [25]. Furthermore, these artifacts can be amplified when the data is sparsely distributed as is often the case for particle filter applications. In an attempt to avoid such artifacts, interpolation based methods are commonly used, however, such methods do not always remove the initial artifacts of the histogram and frequently introduce new ones. Our solution to minimizing histogram-based artifacts is based on an approximation technique referred to as generalized partial volume estimation (GPVE).

GPVE was originally developed for medical and remote sensing image registration applications [25,26,27]. The problem of estimating the joint probability density of two uni-dimensional data sequences, however, provides a slight alteration to the image registration problem. Consequently, in order to apply the basic GPVE principles to the application of estimating joint histograms from 1-D data sequences, the original GPVE histogram estimation procedure outlined in [25,26,27] needs to be reformulated using the following mapping:

$$\begin{aligned} X^* &: \mathrm{x} \rightarrow X^*(\mathrm{x}), \quad \mathrm{x} \in X \\ Y^* &: \mathrm{y} \rightarrow Y^*(\mathrm{y}), \quad \mathrm{y} \in Y \end{aligned} \tag{4.5.1}$$

where X is the discrete domain of the data sequence X^* and Y is the discrete domain of the data sequence Y^*. In addition, the values $X^*(\mathrm{x})$ and $Y^*(\mathrm{y})$ represent the actual data values of the two data sequences at the corresponding indices $\mathrm{x}=x_i$, $\mathrm{y}=y_j$ in terms of the sample spacing. Defining a real-valued kernel satisfying the constraints

$$\begin{aligned} &1. \quad f(n) \geq 0, \quad \text{and} \\ &2. \quad \sum_{n=-\infty}^{\infty} f(n) = 1 \end{aligned} \tag{4.5.2}$$

where $n \in Z$ and Z is the set of all integers, for each $x_i \in X$, $y_j \in Y$, the joint histogram of X^* and Y^* can be updated in the following manner:

$$h(X^*(x_i+p), Y^*(y_j+q)) \leftarrow h(X^*(x_i+p), Y^*(y_j+q)) + f_{X^*}(p) \cdot f_{Y^*}(q) \quad \forall p,q \in Z \,. \tag{4.5.3}$$

From (4.5.3) it can be seen that the increment of the joint histogram is represented in terms of the kernel functions corresponding to each data sequence. Furthermore, the first constraint in (4.5.2) ensures that the increments are non-negative while the second constraint in (4.5.2) ensures that the sum of the updated amounts are normalized to one for each corresponding pair of indices x_i in X^* and y_j in Y^*.

The appropriate selection of a suitable kernel for use in (4.5.3) can depend on many factors, some of which may be the histogram grid parameters or the sparseness of the data in X^* relative to Y^*. Figure 7 illustrates various kernels of the form $f \sim N(0,\sigma^2)$, all of which satisfy (4.5.2).

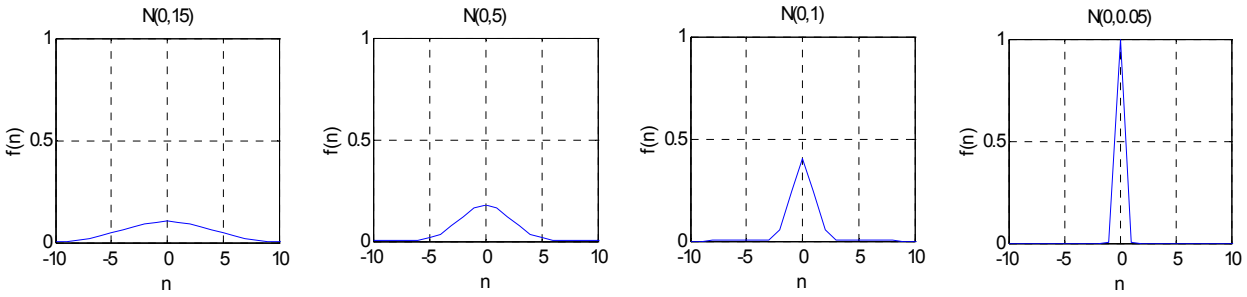


Figure 7: Various GPVE Kernels

Figure 8 illustrates a range-only particle distribution with standard, cubic-interpolated and GPVE histogram estimation – all independently normalized. From this figure it can be seen that the standard histogram generates a significant amount of distortion in the estimated distribution due to the sparseness of the particle data. Consequently, a poor representation for the range-only particle distribution results. The interpolated histogram minimizes some of the distortion observed in the standard approach, however, a substantial amount of distortion still remains. Finally, the GPVE histogram is observed to accurately reflect the true distribution of the range-only particle distribution.

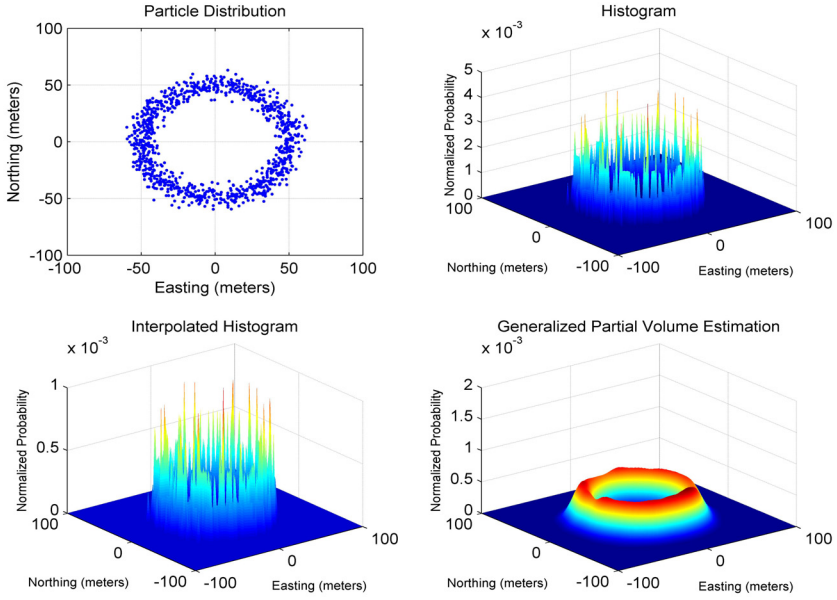


Figure 8: Joint Histogram Estimation of Range-Only Particle Distribution

5.0 PROBABILISTIC DDF – SIMULATED PERFORMANCE RESULTS

Figure 9 illustrates the target path and sensor configuration for a simulated scenario consisting of two targets, three range-only sensors and one bearing-only sensor. In this simulation, each target traverses around the target path counter-clockwise one complete revolution. All local posteriors are estimated using 2000 particles. The sensor likelihood parameters used in (4.2.2) are given as follows:

1. Range-Only Sensors
   $\sigma_{z_r} = 50$ meters downrange error at 500 meters
   $\theta_{min} = 0$ degrees
   $\theta_{max} = 360$ degrees
2. Bearing-Only Sensor
   $\sigma_{z_\theta} = 5$ degrees (90 meters crossrange error at 500 meters)
   $r_{min} = 0$ meters
   $r_{max} = 2000$ meters

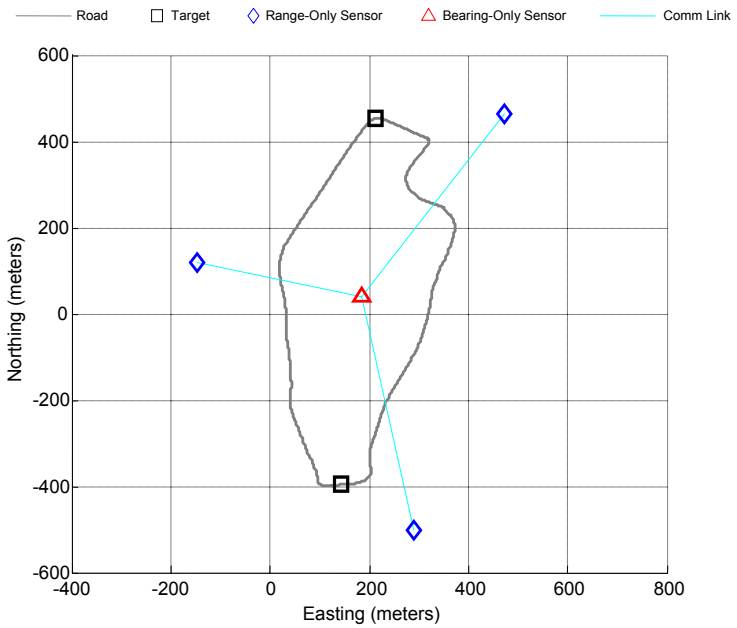


Figure 9: Simulation Scenario

Figure 10 illustrates snapshots of local posterior (top row) and global posterior (bottom row) particle distributions – color-coded relative to a specific sensor - for no fusion, 3 fusions, 6 fusions and 10 fusions. From this figure it can be seen that the fused global posteriors for all sensors quickly collapse to the associated target of interest.

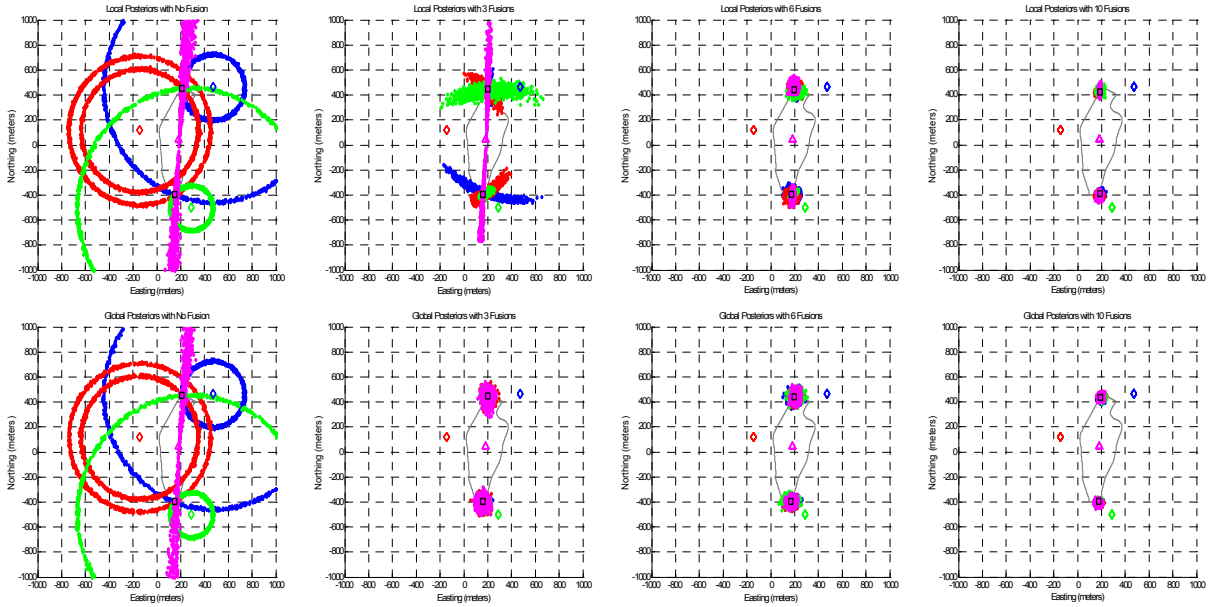


Figure 10: Particle Distribution Snapshots

Figure 11 illustrates the norm of the position error between each target and the mean of the corresponding associated particle set at the bearing-only sensor[4]. From Figure 11 it is interesting to observe that certain regions exist where the sensor modalities are not as complementary as other regions (i.e., approximately 60, 130, 300 and 390 seconds). These regions of concern can be compensated for, however, by either adding more sensors, reconfiguring the sensor modalities or rearranging the sensor placement. Regardless, from Figures 10 and 11 it can be seen that the probabilistic architecture outlined in this work is capable of accurately and efficiently fusing multi-source sensor data in non-linear/non-Gaussian environments.

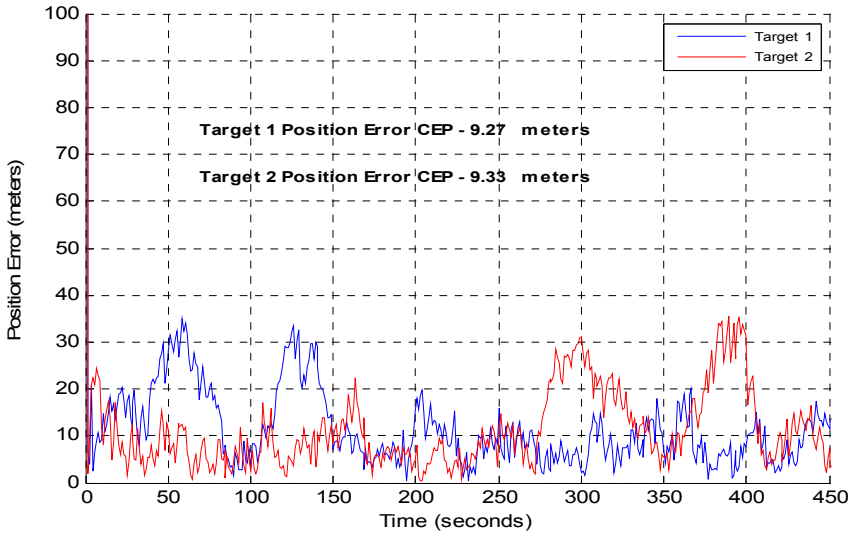


Figure 11: Position Error

[4] The error statistics for the range-only sensors are nearly identical to the error statistics of the bearing-only sensor illustrated in Figure 11.

6.0 SUMMARY AND FUTURE WORK

A highly-scalable, Bayesian approach to the problem of performing multi-source data fusion and target tracking in decentralized sensor networks has been presented. The proposed probabilistic approach provides full decentralization; seamless integration and efficient fusion of multi-source sensor data, and an unprecedented scaling capability. Two primary limitations of the proposed architecture, however, lie in its current inability to parameterize multi-modal distributions for information communication and its lack of a method for handling common information during fusion. In order to maintain efficient scalability, it is believed that methods of density parameterization and fusion should continue to be rooted in techniques utilizing high-order statistics. Consequently, a possible solution for parameterizing multi-modal distributions via high-order statistics could be found through quantile or copula mixtures. In addition, a possible solution for removing common information during fusion could also be found through a quantile or copula based method approaching a similar technique as outlined in [12].

ACKNOWLEDGEMENTS

This paper reports the results of various linear decentralized data fusion implementations designed and developed by BAE Systems Advanced Technology Centre. BAE Systems Advanced Technology Centre would like to thank Andy Cook, Mark Williams, Valerie Leung, Gwynfor Jones, and Chris Smith for their work on the Closed Loop Control demonstration. This paper also reports the results of a decentralized probabilistic data fusion architecture developed by BAE Systems Advanced Systems & Technology under the Advanced Sensors Collaborative Technology (ASCTA) program funded by the Army Research Laboratory through Contract No. DAAD19-01-02-0008. The technical point of contact for this program is Mr. Nino Srour.

REFERENCES

[1] H. Durrant-Whyte and M. Stevens, "Data Fusion in Decentralized Sensing Networks," *Proceedings of the 4th International Conference on Information Fusion*, 7-10 August 2001, Montreal, Canada.

[2] E. Nettleton, "Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles," PhD Thesis, University of Sydney, February 2003.

[3] J. McClellan, G. Edelson and R. Chellappa, "The Listening Eye," *Proceedings of the 2004 Collaborative Technology Alliance Conference*, 5-7 May 2004, Washington D.C.

[4] R. Alexander, J. Anderson, J. Leal, D. Mullin, D. Nicholson and G. Watson, "Distributed Picture Compilation Demonstration," *Proceedings of SPIE (Signal Processing, Sensor Fusion and Target Recognition XIII)*, Vol. 5429, 12-14 April 2004.

[5] D. Dudgeon, G. Edelson, J. McClellan and R. Chellappa, "Listening Eye II," *Proceedings of the 2005 Collaborative Technology Alliance Conference*, 31 May - 3 June 2005, Crystal City, VA.

[6] J. Broussard and M. Richman, "Decentralized Common Operating Picture Compilation in Support of Autonomous Cooperative Behaviors," *Proceedings of the 2005 Collaborative Technology Alliance Conference*, 31 May - 3 June 2005, Crystal City, VA.

[7] M. Rosencrantz, G. Gordon and S Thrun, "Decentralized Sensor Fusion with Distributed Particle Filters," *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 7-10 August 2003, Acapulco, Mexico.

[8] M. Borkar, V. Cevher and J.H. McClellan, "A Monte-Carlo Method for Initializing Distributed Tracking Algorithms" *Proceedings of the 2006 International Conference on Acoustics, Speech and Signal Processing*, 14-19 May 2006, Toulouse, France.

[9] M. Ridley, B. Upcroft, L.L. Ong, S. Kumar and S. Sukkarieh, "Decentralised Data Fusion with Parzen Density Estimates," *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, 14-17 December 2004, Melbourne, Australia.

[10] B. Upcroft, L.L Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh and H. Durrant-Whyte,, "Rich Probabilistic Representations for Bearing Only Decentralised Data Fusion," *Proceedings of the 8th International Conference on Information Fusion*, 25-29 July 2005, Philadelphia PA.

[11] L.L. Ong, M. Ridley, B. Upcroft, S. Kumar, T. Bailey, S. Sukkarieh and H. Durrant-Whyte, "A Comparison of Probabilistic Representations for Decentralised Data Fusion," *Proceedings of the 2005 Intelligent Sensors, Sensor Networks, and Information Processing Conference*, 5-8 December 2005, Melbourne, Australia.

[12] L.L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh and H. Durrant-Whyte, "Decentralized Data Fusion with Particles," *Proceedings of the 2005 Australasian Conference on Robotics and Automation*, 5-7 December 2005, Sydney, Australia.

[13] X. Sheng, Y.Y. Hu and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, pp. 181-188, 25-27 April 2005, Los Angeles, CA.

[14] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 174-188, February 2002.

[15] D. Nicholson, C. Lloyd, S. Julier and J. Uhlmann, "Scalable distributed data fusion," *Proceedings of the 5th International Conference on Information Fusion*, 8-11 July 2002, Annapolis, MD.

[16] M. Ridley, E Nettleton, S. Sukkarieh and H. Durrant-Whyte, "Tracking in Decentralised Air-Ground Sensing Networks," *Proceedings of the 5th International Conference on Information Fusion*, 8-11 July 2002, Annapolis, MD.

[17] S.J. Julier and J.K. Uhlmann, "A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations," *Proceedings of the IEEE American Control Conference*, Vol. 4, pp. 2369-2373, June 1997, Albuquerque, NM.

[18] SIAP SE TF Technical Report 2001-001: Single Integrated Air Picture (SIAP) Attributes. (2001, June). Arlington, VA: SIAP SE TF.

[19] N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar & Signal Processing*, Vol. 140, No. 2, pp. 107-113, April 1993.

[20] M. Orton and W. Fitzgerald, "A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 216-223, February 2002.

[21] R.A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-6, No. 4, pp. 473-483, July 1970.

[22] J.S. Ramberg and B.W. Schmeiser, "An Approximate Method for Generating Asymmetric Random Variables," *Communications of the ACM*, Vol. 17, No. 2, pp. 78-82, February 1974.

[23] E.F. Mykytka and J.S. Ramberg, "Fitting a Distribution to Data Using an Alternative to Moments," *Proceedings of the 11th Winter Simulation Conference*, pp. 362-374, 3-5 December 1979.

[24] J.W. Tukey, Technical Report 36, Statistical Techniques Research Group, Princeton University, 1960.

[25] H. Chen and P.K. Varshney, "Registration of Multimodal Brain Images: Some Experimental Results," *Proceedings of SPIE (Sensor Fusion: Architectures, Algorithms, and Applications VI)*, Vol. 4731, pp. 122-133, 3 April 2002, Orlando, FL.

[26] H. Chen and P.K. Varshney, "Mutual Information Based CT-MR Brain Image Registration Using Generalized Partial Volume Joint Histogram Estimation," *IEEE Transactions On Medical Imaging*, Vol. 22, No. 9, pp. 1111-1119, September 2003.

[27] H. Chen, P.K. Varshney and M.K. Arora, "Performance of Mutual Information Similarity Measure for Registration of Multitemporal Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 41, No. 11, pp. 2445-2454, November 2003.