A Comparison of Probabilistic Representations for Decentralised Data Fusion

# Lee-Ling Ong, Matthew Ridley, Ben Upcroft, Suresh Kumar, Tim Bailey,
Salah Sukkarieh and Hugh Durrant-Whyte

ARC Centre of Excellence in Autonomous Systems (CAS), The University of Sydney, Australia
www.cas.edu.au, s.ong@cas.edu.au

Abstract

This paper compares and constrasts three different probabilistic models for non-Gaussian, non-linear feature tracking, when applied to multiple autonomous vehicles using the Decentralised Data Fusion (DDF) paradigm. Particle representations, Parzen density estimates, and Gaussian mixture models all seek to approximate the probability distributions of an ideal Bayesian filter and have different properties with regard to computational efficiency and quality of the approximation.

In order to satisfy the DDF requirements of modularity, scalability, and robustness, the performance of each representation is simulated for multi-sensor bearing-only tracking simulations. Performance is evaluated in three areas: (a) mathematical accuracy and optimality of fusion for correlated information between nodes, (b) computational efficiency and accuracy of various operations in the DDF framework and (c) bandwidth requirements for communicating the representations over a wireless network.

1. Introduction

This paper aims to provide a comparison of the attributes of non-Gaussian, non-linear probabilistic models for decentralised, multi-sensor tracking particularly for autonomous air and ground platforms. Despite the variety of techniques for non-Gaussian and/or non-linear DDF, no contrasts have been made on the performances of these algorithms. Gaussian mixture models (GMMs) [1], Parzen density estimates [10], and particles [6], three popular representations, will be compared in this paper with emphasis on DDF.

A decentralised sensor network usually comprises of multiple processing nodes supporting one or more sensors. These nodes are interconnected via ethernet or wireless communications. There are three main constraints to a DDF system. The first is that no single central fusion centre exists and no node is central to the operation of the network. Secondly, communications are kept on a strictly node-to-node basis and last, there is no global knowledge of the network topology [5].

Practical applications of DDF have been focused on representing features with simple geometric models such as points, circles or lines with Gaussian noise and through the use of ranging devices such as laser and sonar. While such techniques have been successfully used in autonomous air, ground, and underwater vehicles, constructing accurate models of unstructured and complex environments is difficult. On the other hand, vision sensors return rich feature information such as colour, texture and reflectivity. This information is difficult to model as the appropriate process and observation models are non-Gaussian. Hence, non-geometric probabilistic representations and more general filtering techniques must be considered.

Non-Gaussian probabilistic representations include particle filters, Gaussian mixtures, Parzen density estimates and grid based distributions [16]. Particle filters can be used represent arbitrary distributions but there are currently no known solutions for performing DDF directly on particles. Although DDF can be performed using Parzen and Gaussian mixture representations, the main drawback is the numbered components increase from fusion and update operations requiring the need for reparameterisation. The application of grid-based methods are limited due to the fact they do not scale well with state dimensionality.

In this paper, the performance of particle filters, GMMs, and Parzen density estimates will be contrasted in areas such as computational efficiency of Bayesian estimates, the memory space required to run the filters, the mathematical consistency in which data fusion can be achieved, as well as bandwidth requirements. An application that benefits from this study is decentralised non-linear tracking in unstructured and complex environments using only visual sensory input. In our experiments, we found that the more superior representation is one in which particle filters are used for local filtering and GMMs for decentralised fusion.

2. Related Work

A scalable Gaussian based DDF architecture implemented on UAVs has been successfully demonstrated by Nettleton [9], using a Kalman Filter and its Information form. Local and communicated information was fused asynchronously via additive information matrices. However, this methodology does not lend itself to extensions for general probabilistic distributions.

A problem faced in DDF is the removal of common information. Common information between two nodes from any received estimates needs to be accounted for, if fusion is to be consistent [5] and ensure conservation estimates. It was shown in Utete [19] that there is no finite nor general solution for optimally identifying common information in sensor networks of arbitrary topology. A Covariance Intersect (CI) algorithm [8] provides a means to generate conservative updates between multiple decentralised nodes in a Gaussian-type filter. Otherwise, channel filters [5] may be used in tree structured networks.

Rosencrantz *et al.* [12] and Ihler *et al.* [7] demonstrated DDF using non-Gaussian representations but the guarantee for conservative fusion updates is not considered. Rosencrantz *et al.* decentralised a standard particle filter by communicating and fusing the most informative subsets of samples. Ihler on the other hand, introduced an approximate communication algorithm known as non-parametric belief propagation for non-parametric distributions.

GMMs and Parzen density estimates were first considered by Alspach and Sorenson [15], [1] for target tracking where it was suggested that the use of a single Gaussian kernel with multiple means and weights simplified the target tracking application. Ridley *et al.* [11] showed the use of Parzen density estimates [10] with Gaussian kernels for DDF. The use of Gaussian mixture models in DDF systems was demonstrated by Upcroft *et al.* [18].

Both Upcroft *et al.* and Ridley *et al.* accounted for common information although in a non-optimal sense. Ridley *et al.* developed an analytical approximation to the division of the local estimate and the communicated estimate which was shown to be numerically conservative. Upcroft *et al.* applied a GMM covariance intersect method to remove common information in Gaussian mixture representations. However for both methods, no analytical error bounds on illustrating the consistency were shown.

This paper will try to compare these three representations for DDF while ensuring common information is removed in a numerically consistent way. The rest of the paper is organised as follows: A generalised DDF node is described in Section 3, showing how common information is maintained in a node. Section 4 introduces the form of the models used while the filtering operations such as the local update are discussed in Section 5. The models are then applied to a non-linear, non-Gaussian example in Section 6. Section 7 provides results and discussions on the performances of each representation at decentralised non-linear non-Gaussian tracking. Section 8 concludes and shows future directions.

3. Generalised Decentralised Data Fusion

In a DDF system, each sensor node processes raw sensory data to generate a likelihood. This likelihood is fused with the local estimate of the feature which is then communicated to the other nodes in the network via the channel filters [5]. The internal structure of a decentralised node is illustrated in Figure 1.

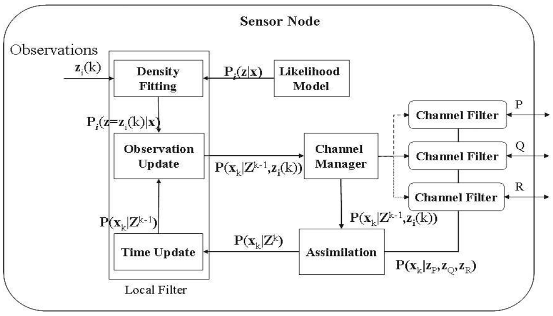


Fig. 1: A Generalised Decentralised Node structure

A. Channel Filters

Channel filters are used for maintaining an estimate of common information passed between two nodes [5]. The removal of common information between the communicated and local estimate is essential in order to avoid over-confident estimates due to "double counting". A channel filter maintains the common information ($P\left(\mathbf{x}_k|\mathbf{Z}_a^k \bigcap \mathbf{Z}_b^k\right)$) between two nodes a and b. Should node b transmit its new state to node a then the Bayesian channel update [2] at node a is:

$$P\left(\mathbf{x}_k|\mathbf{Z}^k\right) \propto \frac{P\left(\mathbf{x}_k|\mathbf{Z}_a^k\right) P\left(\mathbf{x}_k|\mathbf{Z}_b^k\right)}{P\left(\mathbf{x}_k|\mathbf{Z}_a^k \bigcap \mathbf{Z}_b^k\right)} \tag{1}$$

where $P\left(\mathbf{x}_k|\mathbf{Z}^{k-1}\right)$ is the prior belief of the state, $P\left(\mathbf{z}_k|\mathbf{x}^k\right)$ is the probabilistic method for combining observations $\mathbf{Z}^k$ of a state x_k, at time t_k and $P\left(\mathbf{z}_k|\mathbf{Z}^{k-1}\right)$ is the posterior distribution. Equation 1 illustrates that a division operation is required in a channel update for removal of the common information held between communicating nodes. This division is the main problems encountered in generalised DDF.

4. The probabilistic representations

This section provides a description of the mixture of Gaussians filter, the Parzen estimates filter and the particle filter.

A. The Mixture of Gaussians Filter

A Gaussian mixture model for a random variable **X** is $P(x) = \sum_{i=1}^{n} \gamma_i G_i(x)$, where x are the observations of X, G_i, is the probability density for the *i*th component, and γ_i are the weights where $\sum_{i=1}^{n} \gamma_i = 1$.

B. The Parzen Density Estimates Filter

Whilst any type of kernel may be used to represent a Parzen probability distribution, Gaussian kernels are commonly used, as most of its operations are closed in form and therefore more efficient. The Parzen density estimator is the same as a Gaussian mixture model except the covariance of each component is equal and the number of components equal the number of points in the data sample taken from the underlying probability distribution.

C. The Particle Filter

Particle filters are a Monte Carlo estimation method based on importance sampling, adapted to sequential filtering for dynamic systems [6]. The probability distribution of the state, is represented by particles at a given moment in time k, as a set of weighted samples $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, such that the density is approximated by an empirical estimate,

$$P\left(\mathbf{x}_k|\mathbf{Z}^k\right) \approx \sum_{i=1}^{N} w_k^{(i)} \delta\left(\mathbf{x}_k^{(i)}\right) \tag{2}$$

where $\delta(\cdot)$ is the Dirac delta function.

5. Comparing the Local Update, Prediction and Reparameterisation operations

This section addresses the prediction and local update operations. The reparameterisation or resampling operations are described with a comparison of the computational complexity of each option. A linear Gaussian tracking simulation is carried out to demonstrate the validity of these operations.

A. The Mixture of Gaussians Filter

The update operation for a mixture density estimate filter is a multiplication of the two estimates shown as:

$$\begin{aligned} P_{ij}(x) = P_i(x)P_j(x) &= \sum_{i=1}^{n_i} \gamma_i G_i(x) \sum_{j=1}^{n_j} \gamma_j G_j(x) \\ &= \sum_{ij=1}^{n_{ij}} \gamma_{ij} G_{ij}(x), \text{where } n_{ij} = n_i \times n_j \end{aligned} \tag{3}$$

and $G_{ij}(x)$ is the product of two Gaussians ($G_i(x)$ and $G_j(x)$). The solution to the product of two Gaussian mixtures is shown by Sorenson and Alspach [15]. The component increase to size n_{ij} is the resultant of this operation.

If the process model is Gaussian, prediction proceeds in the normal manner with each component of the estimate.

To reduce the number of components, a reparametrisation method is applied. An Expectation-Maximisation (EM) algorithm [4], though computationally expensive may be used to estimate the distribution with a smaller number of parameters. This algorithm has a computational complexity of $O(i \times ND^2)$ where D is the dimensionality of the state, N is the number of components points and i is the number of iterations [4]. Other reparameterisation alternatives include the joining and clustering algorithms introduced by Salmond [13] in which components or groups of components are combined while ensuring the distance between the original distribution remains within a pre-specified error bound. Compared to the EM algorithm, the joining algorithm is far less computationally complex at $O(N \log N)$ where N is the number of components.

B. The Parzen Density Estimates Filter

The local prediction and update operations for a Parzen density estimator are as for a Gaussian mixture model and hence, also results in an increase in the number of the components after an update operation. A subset of Salmond's joining algorithm [13] is applied to reduce the number of components.

C. The Particle Filter

The basic operation of a particle filter, as described in the seminal paper of Gordon *et al.* [6], is to recursively estimate the posterior distribution at the next time-step via a sequence of sampling, importance weighting and resampling.

The update operation results in sample impoverishment or poor exploration of state-space. This is a significant problem as it causes the possibility of obtaining invalid statistics or divergence. The solutions are either performing resampling operations or increasing the sample size. Compared to the reparameterisation step, the resampling step in the particle filters is of order $O(N)$ where N is the number of sample points.

D. Comparison to a linear Gaussian system

In linear Gaussian tracking, the linear Kalman filter is the optimal solution. To demonstrate the ability of each representation to approximate the underlying distribution each model will be compared to linear Kalman filter solution using a chi-squared measure [2]. The observation likelihood of the nine Parzen density components and the five GMM were generated as in section 6-A, with the components spread evenly in the Cartesian-coordinate space around the observation mean. This likelihood is then updated to the local estimate and reparameterisation or resampling is performed.

Figure 2 shows the representations approximate the known underlying distribution accurately. The GMMs and Parzen density estimate provide the best approximation with a maximum error in the 10^{-28}, whereas the Particle representation of 1000 is less accurate (with a maximum error of 0.2), due to the accuracy of obtaining a chi-square measure from particles.

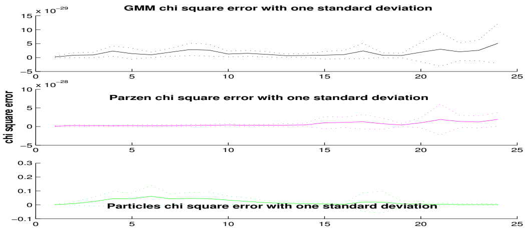


Fig. 2: Chi-square error compared to a Kalman filter

6. Bearing-only Tracking

The Kalman filter often fails [6] in non-Gaussian, non-linear tracking such as bearing-only tracking. These representations will be evaluated in such a senario. In this simulation a feature moved randomly in the x-y plane. The system process model used for prediction was the Integrated Ornstein-Uhlenbeck process [16] which allows for bounding of the Brownian velocity over time. The observations were a sequence of bearing measurements: $z_k = \arctan(\frac{y_k}{x_k}) + v_k$ where z_k is the target bearing and v_k, is the measurement noise.

A. Bearing-only observation likelihood generation

Parzen density and mixture of Gaussian representations require an approximate transformation of the likelihood from the bearing-only measurement-space to a sensor-centric Cartersian-coordinate space. The reason behind this is that their update-step requires a linear transformation between likelihood-space and state-space. Particles on the other hand, use the the bearing-only likelihood density directly as they facilitate non-linear transformations between likelihood-space and state-space.

In our experiments, the approximation for the mixture of Gaussian representation is learnt off-line using the EM algorithm [4]. The initial parameters for EM are equally weighted Gaussians spread evenly over the range on the x-axis. Approximately 50 EM iterations were then used to provide a good fit of the parameters over this bearing-only distribution. For the Parzen density approximation, the kernel window selected is dependant on the number of kernels, range cutoff and standard deviation. The kernel centres were manually selected to equally cover the x-y space. The weights are obtained by evaluating the true distribution at each component and then normalised.

The parameters for the observation model selected was a bearing of zero degree and standard deviation of 5 degrees. The cutoff at 70 m was used so that the Parzen and GMMs could represent a finite distance. It is assumed that the observation lies within the range cutoff. Figure 3 displays the likelihood models given these parameters for the Gaussian mixture, Parzen and particle representations respectively. The particle representation is initialised with 2000 particles in the x-y place. The components of the GMMs all lie approximately at the observed bearing whereas there are less components for the Parzen and particle representations at close range. Hence, mixture of Gaussians representations give better approximations for features at closer range.

B. Comparing the accuracy of transformed likelihood model

To accommodate comparisons among various representations, a divergence or distance measure is required to determine the accuracy of each transformed likelihood model. The selected measure, the Bhattacharyya Coefficient [3] is defined as $\rho(x) \equiv \sum_{u=1}^{m} \sqrt{p_u(x) q_u}$ where q is the represented distribution and p is the true distribution (represented by a fine grid).

The required number of components for each representation was found by seeking the approximation that results in a Bhattacharyya Coefficient of 0.95 or above. Figure 4 shows Bhattacharyya coefficient given the number of components for GMMs and Parzen representations. 15-20 components were required for a Gaussian mixture model while 35-50 components were required for a Parzen density representation. About 2000 components were required for the particle representation. As the transformation of the likelihood is not required for particle representations, the accuracy of representing the given distribution in state space for this representation cannot be determined.

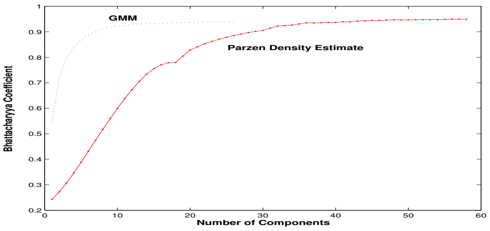


Fig. 4: The bhattacharyya coefficient given number of components for Parzen and mixture of Gaussian models

C. Bearing-only tracking simulation results

Unlike the Gaussian case, the true distribution of this non-linear, non-Gaussian distribution is unknown. However, it can be represented sufficiently with a considerably large number of particles which is used as our 'true' distribution.

Figure 5, shows the accuracy of each representation with the Bhattacharyya coefficient as a measure. Values closer to 1 indicate a solution that is close to the 'true' distribution. It shows all three representations are valid for this example.

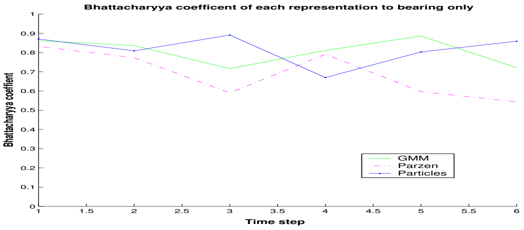


Fig. 5: Bhattacharyya coefficient for each representation

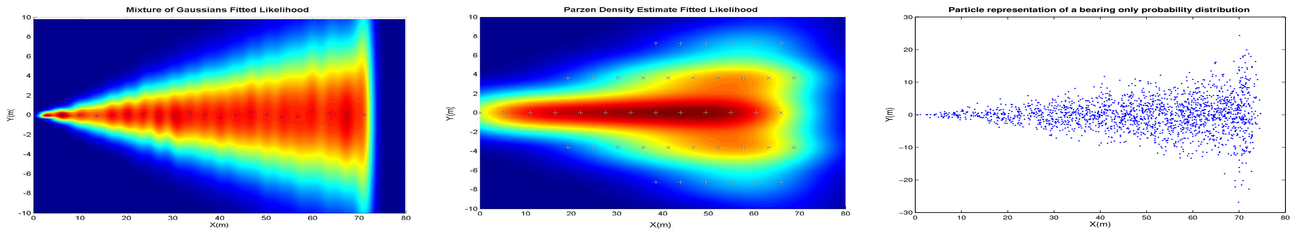
7. Decentralised bearing-only Tracking

Here, the performance of these representations is investigated in a decentralised context including a constrast on the compactness and fusion process of each model. DDF applications require a compact representation particulary for communication as bandwidth is often limited and a consistent fusion process. A feature tracked by two nodes is simulated to demonstrate the accuracy in DDF for each representation.

A. Comparison of the compactness of each representation

Table 1 shows the bandwidth and storage requirements for each representation. The storage requirements are larger than the bandwidth requirements for the Parzen and Gaussian mixture representations as only the upper triangle of the covariance matrix of both estimates need to be communicated while the local estimate consists of the full covariance matrices.

The model to represent the state estimate required 20 GMMs, 50 Parzen components and 2000 particles. With a storage or communication bandwidth of 500 floats, the maximum number of particles is 500 which could prove insufficient, whereas the mixture of Gaussian and Parzen representations would exhibit more superior performances. The Parzen density estimate is the most compact especially for higher dimensions.



(a) A mixture of Gaussians representation

(b) A Parzen representation

(c) A particle representation used for initialisation

Fig. 3: The likelihood representations for a zero degree bearing and standard deviation of 5 degrees, over a range of 70m

TABLE 1: STORAGE AND BANDWIDTH REQUIREMENTS

Represent -ation	Dimen -sion	Components for storage	Components for comms	Bandwidth or storage available
GMM	4	23	33	500
Parzen	4	96	98	500
Particle	4	500	500	500
GMM	6	11	19	500
Parzen	6	66	68	500
Particle	6	500	500	500

For a dimension of 6, the maximum of 11 components that can be communicated may not be as sufficient as 66 Parzen components.

B. Comparing Fusion and Removal of Common Information methods

The fusion operation is described and compared with the focus on the removal of common information.

1) The Decentralised Mixture of Gaussians Filter: Fusion for GMMs is performed via a GMM covariance intersect(CI), update [18], ensuring conservative fusion of possibly correlated information. Illustrated in Equations 4, 5 and 6 are the CI operations where Σ_{ij},μ_{ij} and γ_{ij} are the new covariance, mean and weight of the distribution after fusion between the ith component of the local estimate and the jth component of the communicated estimate. A CI weighting parameter ω is selected to minimise the determinant of the result.

$$\Sigma_{ij}^{-1} = \omega\Sigma_i^{-1} + (1-\omega)\Sigma_j^{-1} \qquad (4)$$

$$\mu_{ij} = \Sigma_{ij}(\omega\Sigma_i^{-1}\mu_i + (1-\omega)\Sigma_j^{-1}\mu_j) \qquad (5)$$

$$\gamma_{ij} = \omega\gamma_i + (1-\omega)\gamma_j \qquad (6)$$

2) The Decentralised Parzen Density Estimates Filter: For a Parzen density estimate, Ridley *et al.* [11] showed and numerically justified an analytical approximation to the division operation described in Section 3. Each component of the communicated estimate is divided by the same kernel. This kernel is an approximation of the previously communicated estimate. The result of this division is then updated or fused with the local estimate via multiplication as with a local update.

3) The Decentralised Particle Filter: There are currently no known solutions for performing DDF directly on particles. As particles are discrete representations, samples from one set do not have the same support on the space as samples from another set. Thus, both the multiplication and division of Equation 1 are not well defined. One possible solution is to convert one set into a continuous distribution such as a mixture of Gaussians and use this to update the importance weights of the particles from the other set. However, common information is not removed in this fusion process. An alternative solution is to convert both sets to continuous distributions and apply one of the DDF techniques from Section 7-B.1 or 7-B.2.

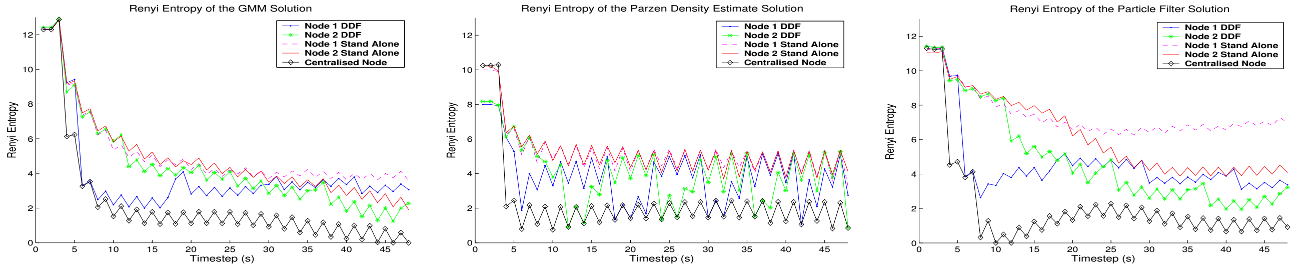
Converting to a mixture of Gaussians representation and then applying CI would be more desirable than a Parzen representation as better summary of the particles can be achieved. The main advantage of selecting particle filters over the other two representations is computational speed. However, converting both sets to continuous distributions would be computationally expensive should communication and fusion occur frequently.

C. DDF bearing-only results

In this simulation, the feature is tracked by two stationary sensing nodes. The process and observation models are the same as in Section 6. In this simulation, prediction occured at every timestep intervals. The local update occured every second timestep while each alternate node communicated its estimate every sixth timestep. The performance comparison used is the optimal centralised solution as it provides the closest approximation to the 'true' solution. Here centralised, means that each node communicating to every other node in the network at every time step. A suitable measure would be one that gives the information content of a distribution such as Entropy [14]. A Quadratic Renyi Entropy suggested by Torkola [17] will be used as the measure between the solution of decentralised or standalone node with the centralised solution.

Figure 6 illustrates the results for each node performing DDF and the standalone nodes (i.e. no communications). The results indicate that decentralised nodes exhibit performances better than the sensors operating alone for each of the three representations. The final solutions for the decentralised nodes are similar but less compact that the centralised one.

More process noise was required for the Particle and Parzen simulation compared to the others because of sample improvishment, hence reducing the accuracy of this representation if the update or fusion processes do not occur frequently.



(a) GMM Renyi Entorpy Solution

(b) Parzen Renyi Entorpy Solution

(c) Particle filter Renyi Entorpy Solution

Fig. 6: Renyi Entropy for the GMMs, Parzen Density Estimates and the Particle filter solutions. It is shown that nodes with decentralised fusion give superior results.

The CI update for the mixture of Gaussians is shown to be more conservative than the decentralised fusion process for the Parzen density estimate as it can be seen in Figures 6(a) and 6(b) that the Renyi Entropy is closer to the centralised solution when fusion occurs compared to the mixture of Gaussians.

8. Conclusions and Future Work

In this paper, it is demonstrated that Particle filters, Gaussian mixture models and Parzen density estimates can satisfy all the constraints for a general Decentralised data fusion architecture and also provide solutions to the operations of general Bayesian filtering.

The best representation for non-Gaussian, nonlinear feature tracking in decentralised data fusion is dependant on the application itself. For this simulation, using the particle representation for a local filter and using mixtures as a summary of the sample statistics for DDF is likely to be the best option. Particle representations do not require an observation likelihood transformation to state space, nor reparameterisation which may be computationally expensive compared to resampling. Parzen density estimates and Mixture of Gaussians are more compact than a subset of particles, can be used for DDF operations. Mixture of Gaussians are selected over Parzen density estimates as the density estimation method used for GMMs is more reliable than the Parzen representation.

One of the major areas of further research is improving density estimation techniques for Parzen estimates and mixture of Gaussians representations. Another is the development of different fusion methods for particle representations. Major advances in data association for multiple target tracking is expected through this work. Future work will also include a demonstration of each of these representations using vision sensors on airborne vehicles, ground vehicles and stationary ground nodes.

Acknowledgments

This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government and is supported by BAE Systems, Bristol, UK.

References

[1] D.L. Alspach, H.W. Sorenson, "Nonlinear bayesian estimation using Gaussian sum approximations", *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, August 1972, pp. 439-448.

[2] Y. Bar-Shalom, Ed., "Multitarget-multisensor tracking: advanced applications", Artech House, 1990.

[3] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, 2003, pp. 564-575.

[4] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society B*, Vol. 39, 1977, pp. 1-37.

[5] S. Grime, "Communication in Decentralised Sensing Architectures" *phD Thesis*, The University of Oxford, 1992.

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

[7] A.T. Ihler, E.B. Sudderth, W.T. Freeman, A.S. Willsky, "Efficient Multiscale Sampling from Products of Gaussian Mixtures", *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, 2004.

[8] S. Julier, J, Uhlmann, "A nondivergent estimation algorithm in the presence of unknown correlations", *In Proceedings of The American Control Conference*, Vol. 4, IEEE, July, 1997, pp. 2369-2373.

[9] E. Nettleton, H. Durrant-Whyte, S. Sukkarieh, "A Robust Architecture for Decentralised Data Fusion", *The 11th International Conference on Advanced Robotics (ICAR'03)*, Coimbra, Portugal, 2003.

[10] E. Parzen, "On estimation of a probability density function and mode", *The Annals of Mathematical Statistics*, Vol. 33, No. 3, September, 1962, pp. 1065-1076.

[11] M. Ridley, B. Upcroft, L.L. Ong, S. Kumar, S. Sukkarieh, "Decentralised Data Fusion with Parzen Density Estimates", *International Conference on Intelligent Sensors, Sensor Networks and Information Processing 2004 (ISSNIP '04)*, Melbourne, Australia, 2004.

[12] M. Rosencrantz, G. Gordon, S. Thrun, "Decentralised Data Fusion with distributed particle filters", *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.

[13] D.J. Salmond, D.P. Atherton, J.A. Bather, "Mixture reduction algorithms for uncertain tracking", *IFAC Proceedings Series*, Vol. 2, No. 8, 1989, pp. 775-780.

[14] C.E. Shannon, "A mathematical theory of communication", *Bell System Technical Jornal*, Vol. 2,7, July and October 1948, pp. 379-423,623-656.

[15] H.W. Sorenson, D.L. Alspach, "Recursive bayesian estimation using Gaussian sums", *Automatica*, Vol. 7, 1971, pp. 465-479.

[16] L.D. Stone, C.A. Barlow, T.L. Corwin, "Bayesian Multiple Target Tracking", *Artech House*, 1999.

[17] K. Torkkola, "Feature extraction by non parametric mutual information maximization", *The Journal of Machine Learning Research*, Vol. 3, March 2003, pp. 1415-1438.

[18] B. Upcroft, S. Kumar, L.L. Ong, M. Ridley, T. Bailey, S. Sukkarieh, H. Durrant-Whyte, "Rich Probabilitstic Representation for Bearing Only Decentralised Data Fusion", *accepted to IEEE Conference on Information Fusion*, 2005.

[19] S. Utete, H. Durrant-Whyte, "Reliability in decentralised data fusion networks", *In Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1994, pp. 215-221.