

Consistent methods for Decentralised Data Fusion using Particle Filters

Lee-Ling Ong, Ben Ucroft, Matthew Ridley, Tim Bailey, Salah Sukkarieh and Hugh Durrant-Whyte

ARC Centre of Excellence in Autonomous Systems (CAS)

The University of Sydney, NSW 2006, Australia

www.cas.edu.au, s.ong@cas.edu.au

Abstract—This paper presents two solutions for performing decentralised particle filtering in view of non-linear, non-Gaussian tracking in sensor networks. The issue is that no known methods exist to deal with correlated estimation errors due to common past information between two discrete particle sets.

The first method transforms the particles to a Gaussian Mixture Model, the second approximates the set by a Parzen density estimate. Both of these representations accommodate consistent fusion and maintain accurate summaries of the particles. Requiring less bandwidth than particle representations, transformations to GMMs or Parzen representations for communication provide an added advantage.

The accuracy in which the algorithms summarise the particle set, fusion methods and bandwidth requirements of each representation will be compared. Our results show that whilst less GMM components are required to summarise the sample statistics, the decentralised fusion solution using Parzen representations yields a more accurate result.

I. INTRODUCTION

Two methodologies for performing decentralised particle filtering are proposed in view of non-linear, non-Gaussian tracking applications in sensor networks. Our application aims to demonstrate decentralised data fusion (DDF) techniques using vision sensors only. Although vision sensors return rich feature information such as colour, texture and reflectivity, the observation model is bearing-only and cannot be modelled as a Gaussian. Hence, non-Gaussian representations are required, in which particle filters are an ideal choice. Other representations considered include grid-based techniques [1], Gaussian Mixture Models (GMMs) [2] and Parzen representations [3]. Grid based representations are not compact and do not scale well with dimension compared to particles. GMMs and Parzen representations require an approximate observation likelihood transformation from the sensor space to Cartesian space and result in a multiplicative increase of parameters at every local update.

The focus of this work is on applications in macro sensor networks, which are characterized by platform mobility, heterogeneous teams and long mission duration. The DDF system under consideration includes mobile autonomous robots, stationary sensor platforms and human operators. The characteristics of DDF considered here are that no node is central to network operation, that no common communication facility exists and no node has global knowledge of the network topology. The advantages imposed by these three constraints include modularity, scalability, survivability and increased robustness. As nodes in a DDF system operate

independently and communicate locally, an operating failure in a node would not affect integrity of the operation of surviving nodes.

In order to perform DDF consistently, new information has to be recovered from the received estimate by removing common past information [4]. Figure 1 shows how common information between two nodes i and j ($P(\mathbf{x}|\mathbf{Z}_i \cap \mathbf{Z}_j)$) arises where \mathbf{x} is the state of vector and \mathbf{Z} is the observation.

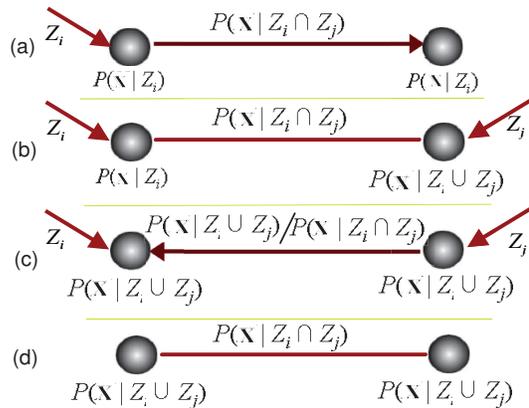


Fig. 1. How common information arises from the network is shown. Communication of estimates (indicated by arrows) are from Node i to Node j at (a) and vice versa at (c). At (c), the estimate communicated contains past information sent from Node i at (a). This common information has to be removed by division so only “new” information is fused at Node i .

In Figure 1, (a) shows that at first instance, Node i makes an observation Z_i , of a feature, updates the local filter resulting in a posterior $P(\mathbf{x}|\mathbf{Z}_i)$, and sends its estimate to Node j , which instantiates the filter for this target. This communicated estimate $P(\mathbf{x}|\mathbf{Z}_i)$ also becomes the common information $P(\mathbf{x}|\mathbf{Z}_i \cap \mathbf{Z}_j)$, between these two nodes.

At (b), New observations are updated at Nodes i and j .

At (c), Node j now sends to Node i , its estimate $P(\mathbf{x}|\mathbf{Z}_i \cup \mathbf{Z}_j)$. This is a combination of information from Node i sent at time (a) and information from local updates at Node j . To avoid errors arising from correlations, the common past information has to be removed from this estimate prior to fusion through a division.

(d) is after the fusion update, where the estimate at both Nodes i and j are based on information from both nodes. The common information is now what Node j communicated at step (c).

Hence, the fusion between two nodes amounts to a division operation to obtain only “new” information from the communicated estimate and a multiplication of this “new” information with the local estimate [5], which is:

$$P(\mathbf{x}|\mathbf{Z}_i \cup \mathbf{Z}_j) \propto \frac{P(\mathbf{x}|\mathbf{Z}_i) P(\mathbf{x}|\mathbf{Z}_j)}{P(\mathbf{x}|\mathbf{Z}_i \cap \mathbf{Z}_j)} \quad (1)$$

The division operation is the main problem encountered in non-Gaussian DDF. The aim here is to develop a mathematically consistent and tractable formulation of the division when using particle filters.

II. PROBLEM STATEMENT

The division operation can be performed analytically with Gaussian representations for tree-connected networks. However, if the correlation between the estimates to be fused is unknown, a covariance intersect filter can be applied [4].

As our selected representation is particle filters, the problem that occurs is consistently fusing two particle sets directly. There is only support for an infinitesimally small interval at the particle. Elsewhere the value of the particle is zero. Hence, there is no overlap between samples even in the same space. Unless two samples lie exactly at the same spot, multiplication between the two sample sets would result in zero as shown in Figure 2. For example, the multiplication of the first sample of set 1 with the first sample in set 2 is $\delta(x - 0.5)\delta(x - 1) = 0$.

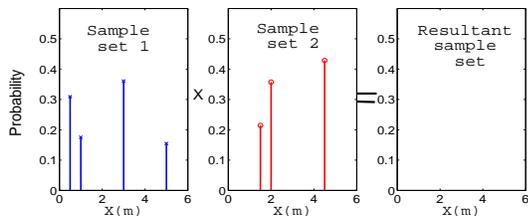


Fig. 2. Samples from one particle set do not have the same support on the space as samples from another set. The particle at 0.5 m from Set 1 will be multiplied by a value of zero at 0.5 m from Set 2 because there is no support at that point at Set 2.

Rosencrantz *et al.* [6] performed fusion on particle filters by adding the most informative subset of samples from two nodes together. This method is mathematically inconsistent with Equation 1 and common past information is not accounted for as a division operation cannot be performed on discrete samples for the same reasons as multiplication.

The main contribution of this paper is the development of two methods for consistent fusion and hence, decentralising particle filters. Both methods require a transformation to continuous representations, one being GMMs and the other Parzen representations. Not only do these continuous representations satisfy conservative fusion update [7], [8], the transformations also smooth and regularise the sample set [9], allowing the particles to be distributed more evenly. The secondary contribution of this paper is the reduction of bandwidth per communication step by communicating the transformed GMMs and Parzen representations instead

of particles. Our simulation results show that less GMM components are required to summarise the sample statistics compared to Parzen components. Additionally, the accuracy of the approximation by GMMs is better than Parzen representations. However, the fusion algorithm for Parzen density estimates is more accurate than the algorithm for GMMs offsetting the benefits of GMMs.

The paper is organised as follows: After presenting some related work (Section III), a generalised DDF node is described in Section IV. Section V introduces the two methods of performing consistent DDF on particle filters. In Section VI, simulation results are presented. Section VII concludes and presents future directions.

III. RELATED WORK

Since the seminal paper by Gordon *et al.* [10], particle filters have been used widely especially in a centralised fashion [11],[12].

Distributive particle filter algorithms were used by Coates [13] and Sheng *et al.* [14] to communicate measurements from low-cost sensors in sensor networks. Coates required that all the nodes maintain particle filters that are synchronized using the same set of prior distribution. The data is encoded as partial likelihood functions prior to propagation throughout the network. The final importance distribution is then back-propagated to all the sensor nodes and a new set of particles will be generated at each sensor using the final distribution. Sheng *et al.*, use an Expected Maximisation (EM) algorithm to train a Gaussian Mixture approximation to the particle representation and communicated the Gaussian Mixture to conserve bandwidth. Communication of observation information when received is not robust to communication failure. Our application communicates the state estimates at constant time instead.

Ihler *et al.* [15] applied a message-passing estimation technique known as non-parametric belief propagation based on a generalisation of particle filtering in sensor networks. These messages are estimates of the location and uncertainty of the sensor nodes themselves, represented as either samples or analytical functions. A Gaussian mixture estimate is computed from the outgoing message for communication. Our application aims not to communicate and represent the sensor estimates but rather features of interest in the environment.

Challa *et al.* [16] used Support Vector Machines (SVM) in distributed data fusion to compress the particle representation of the state estimate for broadcast to a central fusion centre. Rosencrantz *et al.* [6] decentralised a standard particle filter by fusing the most informative subsets of samples without producing a functional distribution over the individual representations. Although Rosencrantz *et al.* and Challa *et al.* used particle filters in a decentralised sense, the guarantee for conservative fusion updates is not considered as common information is not accounted for.

IV. DECENTRALISED NODE STRUCTURE

The operations in a decentralised node is illustrated in Figure 3.

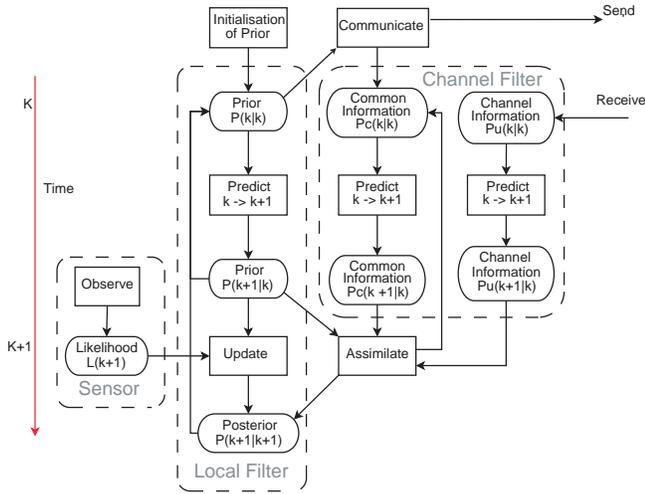


Fig. 3. Flow chart of the operations performed in DDF is shown. The local filter is where the observation likelihoods are updated from local sensors and information received from channel filters is used. The channel filter is used to maintain a record of common information between two nodes.

In a DDF system, likelihoods are generated from observations made by the sensor. Data association is then performed between the likelihood and existing local tracks where either fusion or track initialisation takes place. A standard cycle of a local observation update (multiplication of prior and likelihood) and prediction (convolution of prior with process model) occurs at the node. At set times, the local particle sets are then transformed into a representation that requires less communication bandwidth and communicated to neighbouring nodes in the network via the channel filters [17].

The channel filter also receives information from neighbouring nodes. When this occurs, data association is performed. If associated to a track, the received information is fused after common past information is removed. The common information at the channel filter is also updated with the received information.

V. DECENTRALISED PARTICLE FUSION ALGORITHMS

Particle filters are a Monte Carlo estimation method based on importance sampling, adapted to sequential filtering for dynamic systems [10]. 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,

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

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

As explained in Section II, the fusion operations (Equation 1) of two particles sets cannot be performed directly. At least one set of particles has to be transformed to a continuous distribution to be sampled by the second set to obtain the new importance weights. The continuous distributions considered for transformation are Gaussian Mixture Models (GMMs) and Parzen Density Estimates.

A. The Continuous Distributions

1) *Gaussian Mixture Models (GMMs)*: A Gaussian mixture model for a random variable \mathbf{x} is:

$$P(x) = \sum_{i=1}^n \gamma_i G_i(x; \mu_i, \Sigma_i) \quad (3)$$

where x is in the domain of \mathbf{x} , G_i , is the i th Gaussian component, and γ_i are the weights where $\sum_{i=1}^n \gamma_i = 1$. The multivariate Gaussian distribution of the state x with mean μ and covariance Σ is defined as:

$$P(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp^{-\frac{1}{2} [\mathbf{x} - \mu]^T \Sigma^{-1} [\mathbf{x} - \mu]} \quad (4)$$

2) *Parzen Density Estimates*: For Parzen density estimates, any type of kernel may be used to represent a probability distribution. However, Gaussian kernels are preferred, as the operations are closed form and therefore efficient. The Parzen density estimator is similar to a GMM except each component has the same covariance. The equation for a Parzen density estimate with a Gaussian kernel is:

$$P(x) = \sum_{i=1}^n \gamma_i G(x; \mu_i, \Sigma) \quad (5)$$

where $G(x)$ is the Gaussian probability density on \mathbf{x} and γ_i are the weights where $\sum_{i=1}^n \gamma_i = 1$.

B. Conversion to a Continuous Distribution

The method of converting to a continuous distribution (shown in Figure 4) is based on Musso [9] where each sample is converted to a kernel $K_h(\mathbf{x})$:

$$K_h(\mathbf{x}) = h^D K(\mathbf{x}) \quad (6)$$

where D is the number of dimensions, $K(\cdot)$ is the rescaled kernel density and $h > 0$ is the window or scaling parameter. The kernel selected is Gaussian with

$$h = \left(\frac{4}{D+2}\right)^e N^{-e} \quad (7)$$

where $e = \frac{1}{D+4}$, and N is the number of samples.

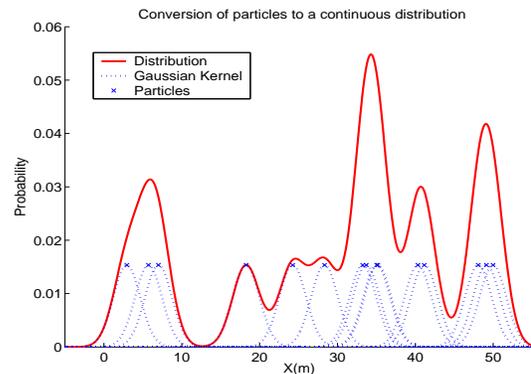


Fig. 4. Conversion of particles to a continuous distribution (shown as the solid line) by placing Gaussian kernels (shown as the dotted lines) over each weighted particle (shown by a \times).

Communicating the continuous distribution in this form would be slightly worse than communicating the sample set itself as there is a kernel for each particle. Hence, approximating this distribution by a more compact one such as Parzen representations or GMMs is more desirable.

1) *Algorithm 1 : Conversion to GMMs:* Expectation Maximisation (EM) algorithm [18] or via West's joining algorithm [19] are two methods considered for reducing the sample set to a more compact GMM.

The EM algorithm finds the maximum likelihood parameter estimation in statistical models with variables that are not observed, given initial parameters. These parameters are determined using the X-means algorithm [20]. The computational complexity for X-means is $O(i \times N \log k_{max})$ where k_{max} is the selected maximum number of cluster means, N is the number of samples and i is the number of iterations. The computational complexity for the EM algorithm for GMMs is $O(i \times ND^2)$ where D is the dimensionality of the state. In our experiments, we find that the EM algorithm requires a large set of at least 2000 particles to perform adequately. We also find that EM is considerably more computationally expensive compared to the West's joining algorithm which has a complexity of $O(M \log M)$ where $M \ll N$ is the number of components. Hence, we chose to implement the latter for reducing the Gaussian sum to a distribution of less components.

The operation of West's joining algorithm is to merge pairs of components from the sum of Gaussian kernels (Equation 5) converted from the particles, successively until the desired level of reduction has been achieved. The distance measure utilized to gauge the similarity of component i and component j of the GMM is a Mahalanobis-type distance measure:

$$d_{ij}^2 = \frac{\gamma_i \gamma_j}{\gamma_i + \gamma_j} (\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j) \quad (8)$$

where μ is the state vector of the component, Σ is the mixture covariance matrix and γ is the component weight.

2) *Algorithm 2 : Conversion to Parzen density estimates:* A variant of West's joining algorithm [19] for Parzen estimates, can be applied to the sums of Gaussian kernels that were converted from the particles (Equation 5). Here, the kernel covariance is adjusted such that the ensemble covariance is preserved after the number of components have been reduced.

C. Fusion and Accounting of Common Information

1) *Algorithm 1 : Fusion via GMMs:* Fusion is performed by first converting the local particle set to a GMM. A generalised covariance intersect (CI) update [7] is then performed with the communicated GMM to ensure conservative fusion of possibly correlated information. Illustrated in Equations 9,10 and 11 are the CI operations where Σ_{ij} , μ_{ij} and γ_{ij} are the new covariance, mean, and weight of the component after fusion between the i th component of the local estimate ($P_a(x)$) and the j th component of the communicated

estimate($P_b(x)$)).

$$\Sigma_{ij}^{-1} = \omega \Sigma_i^{-1} + (1 - \omega) \Sigma_j^{-1} \quad (9)$$

$$\mu_{ij} = \Sigma_{ij} (\omega \Sigma_i^{-1} \mu_i + (1 - \omega) \Sigma_j^{-1} \mu_j) \quad (10)$$

$$\gamma_{ij} = \frac{\gamma_i^\omega \gamma_j^{1-\omega}}{\sum_{k=1}^{N_a} \sum_{l=1}^{N_b} \gamma_k^\omega \gamma_l^{1-\omega}} \quad (11)$$

A CI weighting parameter ω is selected such that the Chernoff Information of the resultant distribution is equally distance from both prior distributions. The optimal ω hence, would have the property of

$$D(P_\omega(x) || P_a(x)) = D(P_\omega(x) || P_b(x))$$

where $D(\cdot || \cdot)$ is the Kullback-Leibler(KL) divergence [21]. The new particle set is then obtained by sampling from the fused GMM.

2) *Algorithm 2 : Fusion via Parzen density estimates:*

A particle set, summarised and communicated as a Parzen density estimate enables an approximate Bayesian channel update operation (a division operation) to be performed at the receiving node to remove common information [8].

In this operation, each Parzen component of the communicated estimate is divided by a Gaussian with the same covariance. A division of a Gaussian by a Gaussian is possible. The covariance is approximated from the previously communicated estimate. This results in all the quotients having the same covariance which maintains form. The mean and weight of the Gaussian divisor is approximated by evaluating the Gaussian divisor and its gradient at the point of interest. This point of interest is the mean of the current Parzen component.

VI. IMPLEMENTATION

The decentralised particle filter algorithms are implemented in a six degree-of-freedom multiple vehicle simulation. In the simulation, two flight vehicles equipped with vision sensor undergo part of a figure of eight trajectory approximately 100 m above the ground with average flight speeds of 144 m/s. The field of view (FOV) angle of the sensors are $\pm 15^\circ$ with bearing uncertainties of 0.1601° and elevation uncertainties of 0.1206° . 50 features were scattered on the ground to be tracked.

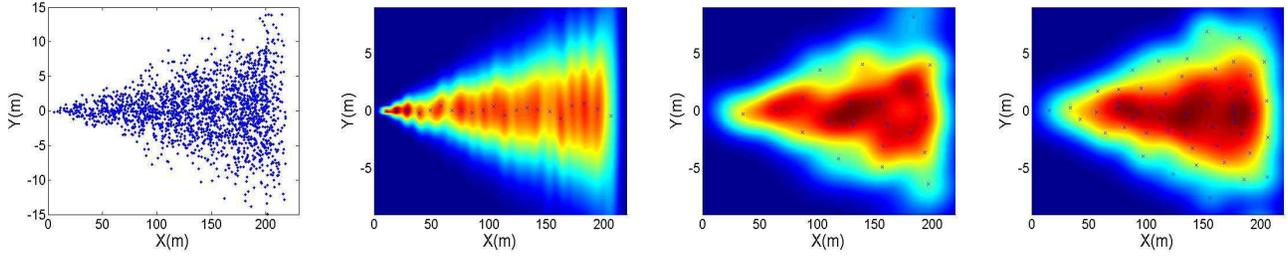
In each local filter, the estimation model is updated every 0.025s, and the observation model is updated every 0.2s. The range-cutoff used when initialising a new filter is 200 m. Every 0.6 seconds, each alternate platform communicates summaries of each sample set.

The system process model used for prediction was the Integrated Ornstein-Uhlenbeck process [1] which allows for bounding of the Brownian velocity over time. This prevents excessively large velocities that can occur due to wayward measurements when the feature is not observed for an extended period.

The observations (\mathbf{z}_k) are a sequence of bearing (φ) and elevation (ϑ) measurements:

$$\mathbf{z}_k = [\varphi \ \vartheta]^T = \begin{bmatrix} \tan^{-1}(y_k/x_k) \\ \tan^{-1}(z_k/\sqrt{x_k^2 + y_k^2}) \end{bmatrix} + \mathbf{v}_k \quad (12)$$

where \mathbf{v}_k is the measurement noise.



(a) A bearing-only probability distribution with 2000 particles (b) A GMM approximation of the particle representation via EM compromising of 23 components (c) A GMM approximation of the particle representation via West's joining algorithm compromising of 23 components (d) A Parzen approximation of the particle representation via West's joining algorithm compromising of 55 components

Fig. 5. The GMM and Parzen approximations (5(b) to 5(d)) of a bearing-only probability distribution 5(a). The range-cutoff for this distribution is at 200 meters, the bearing is zero degrees with a variance of 1.5 degrees

TABLE I
BANDWIDTH REQUIREMENTS

Representation	Dimension	Components for comms	Bandwidth available (1 MTU)	Data Transmitted
				No. of Components [No of bytes] parameter
GMM approx	4	97	1460	97[4]means, 97[10]covariance, 97[1]weights
Parzen approx	4	290	1460	290[4]means, 1[10]covariance, 290[1]weights
Particle set	4	365	1460	365[4]means
GMM approx	6	52	1460	52[6]means, 52[21]covariance, 52[1]weights
Parzen approx	6	205	1460	205[6]means, 1[21]covariance, 205[1]weights
Particle set	6	243	1460	243[6]means

A. Accuracy of each particle summary

The accuracy of each representation in summarising the particle set is shown. The particle set shown in Figure 5(a) is a set of 2000 particles representing a range-limited bearing only probability distribution. The range cutoff is set at 200 metres for a bearing at zero degrees and a variance of 1.5 degrees.

The transformations of this sample set to a GMM using EM and West's joining are shown in Figures 5(b) and 5(c) and the transformation to a Parzen in 5(d). Note that the EM result of this transformation is more cone-shaped, particularly between the ranges of 0 and 50 metres.

To accommodate comparisons among various representations, a divergence or distance measure is required to determine the accuracy of each approximation. The selected measure, the Bhattacharyya Coefficient [22] is defined as

$$\rho(x) \equiv \sum_{i=1}^m \sqrt{p_i(x)q_i(x)} \quad (13)$$

where q is the represented distribution and p is the true distribution (represented by a fine grid). The minimum number of components was found by seeking the approximation that resulted in a coefficient of 0.95 where a value of 1 would indicate that $p = q$.

In this example, for the GMM approximation, a reduction to 23 components resulted in a Bhattacharyya coefficient of 0.95. To obtain the same coefficient, 55 Parzen components were required. Hence, less components are required for a GMM approximation compared to a Parzen window approximation. 23 Parzen components would result in a less accurate

coefficient of 0.92. However, the approximation to 23 GMMs via the EM algorithm is more accurate with a coefficient of 0.99 but computationally expensive.

B. Bandwidth requirements

In sensor networks, it is desirable to communicate a representation as compact as possible. Table I shows the bandwidth requirements for communicating :

- (a) the particle set directly
- (b) a Parzen density approximation of the particles and
- (c) a GMM approximation of the particles

Two examples are provided, which represents state dimensions of 4 and 6. Only the upper triangle of the symmetric covariance matrix of the Parzen and GMM representation need to be communicated. For example, the number of bytes to store the covariance matrix for a state dimension of 4 is 16 (4×4). Communicating the upper triangle of this matrix would require 10 bytes. In our example, the data has to be communicated in one MTU (Maximum Transmission Unit) which is 1500 bytes. An MTU is the maximum size of a packet that can be transmitted in one frame over a network. As a number of bytes are allocated to the source and destination IP addresses of the packet, the actual communicated data size is limited to 1460 bytes.

For a 4-D system, 23 GMM or 55 Parzen components were required to approximate a particle set of 2000. With a communication bandwidth of 1460 floats per time interval, the maximum number of particles is 365 which could prove insufficient, whereas the GMM and Parzen approximations would exhibit better performances as 4 GMM feature es-

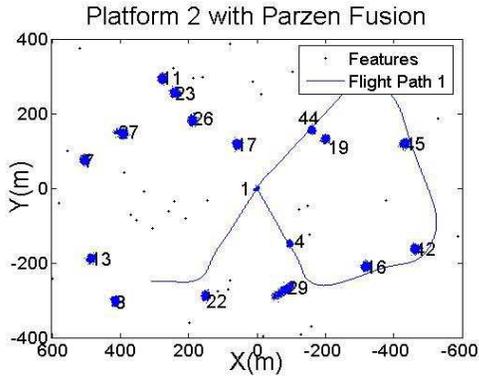


Fig. 6. Parzen Fusion results from Node 2 - 17 features tracked

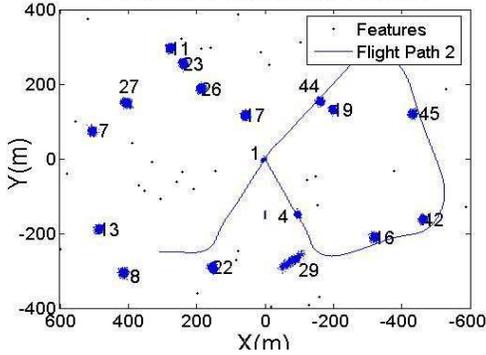


Fig. 7. GMM Fusion results for Node 2 - 17 features tracked

timates and 5 Parzen estimates can be communicated. The Parzen density estimate is the most compact especially for higher dimensions. For a dimension of 6, the amount of data that can be communicated reduces to 52 GMM components which is about 50 % while the number of Parzen components reduces by only 30 %.

C. Results

The final results of the simulation is shown from Figures 6 to 9. In this simulation, features on the ground are observed either by platform 1 alone, platform 2 alone or by both platform 1 and 2. Observations from both platforms were communicated to obtain the centralised solution. For the GMM and Parzen fusion solution, the estimates were communicated at a lesser frequency. Hence, more tracks are made even though the feature was not observed by the platform itself such as Features 4 and 29 for platform 1, increasing the area of coverage for that platform. Although the fusion results is not optimal, they matches very closely to the centralised solution. Features 1, 19, 22 and 44 were observed by both platforms. The GMM fusion results for these features are slightly less tightly distributed or less compact compared to the Parzen fusion results.

The results of each fusion method is compared to the centralised solution. The measure used is the relative entropy or Kullback-Leibler(KL) divergence [21] as it measures the inefficiency of each distribution assuming that the centralised solution is the most optimal.

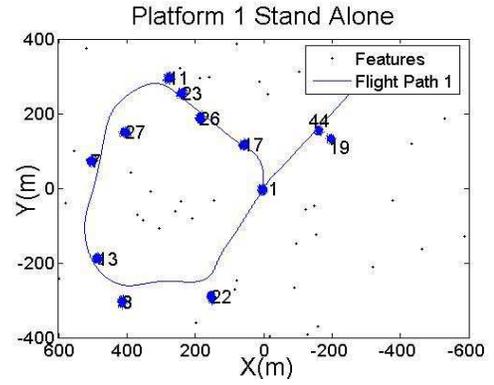


Fig. 8. Platform 1 StandAlone Result - 12 features tracked

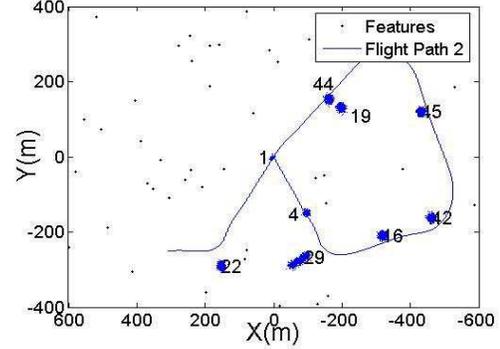


Fig. 9. Platform 2 StandAlone Result - 9 features tracked

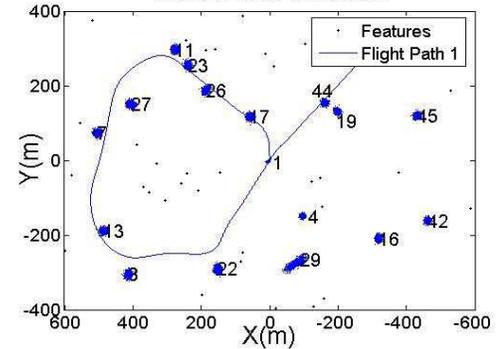


Fig. 10. Centralised Solution - 17 features tracked

Figures 11 and 12 illustrate the KL-Divergence results for each node performing DDF and the standalone nodes (i.e. no communication) compared to the centralised solution for Feature 1. Platform 1 observed the feature at the time of 0-6 s and 166-183 s. Platform 2 observed the feature between the 46-63 s and the 230-240 s. The results show that when one node is observing the feature, communicating and fusing the feature estimate to the other node via Parzen representations is more accurate compared to via GMMs in terms of the KL-Divergence. These results also indicate that decentralised nodes exhibit performances better than the sensors operating alone and that the final solutions for the decentralised nodes are similar but less compact that the centralised one.

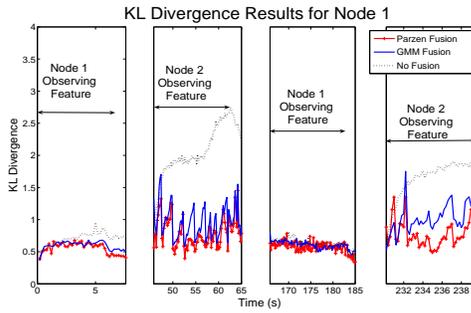


Fig. 11. KL Divergence results of Feature 1 for the Stand Alone Node 1, Node 1 communicating and fusing GMMs, Node 1 communicating and fusing Parzen representations.

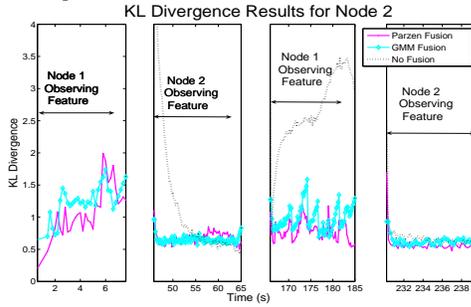


Fig. 12. KL Divergence results of Feature 1 for the Stand Alone Node 2, Node 2 communicating and fusing GMMs, Node 2 communicating and fusing Parzen representations.

VII. CONCLUSION

This paper has introduced two methodologies for performing consistent and efficient decentralised data fusion with particle filters which transform the particle set to either GMMs or Parzen estimates for communication and inter-nodal fusion. This is due to the GMMs and Parzen representations which have the capacity to account for common information between communicated and local distributions.

Summarising the sample set with a GMM requires less components and is more accurate than approximation by Parzen representation. However, the Parzen density estimate is more compact. Better fusion results are obtained using Parzen representations as the Parzen estimate division is more accurate than the generalised covariance intersect for GMMs. However, fusion via the GMM covariance intersect can be performed for arbitrary connection topologies, where the division of Parzens cannot.

One of the areas for future work is the development of different fusion methods for GMMs and particle representations. 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, ATC, UK.

REFERENCES

- [1] L. Stone, C. Barlow, and T. Corwin, *Bayesian Multiple Target Tracking*. Artech House, 1999.
- [2] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17(4), pp. 439–448, August 1972.
- [3] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33(3), pp. 1065–1076, September, 1962.
- [4] S. Julier and J. Uhlmann, "A nondivergent estimation algorithm in the presence of unknown correlations," *In Proceedings of The American Control Conference*, vol. 4, pp. 2369–2373, July, 1997.
- [5] Y. E. Bar-Shalom, *Multitarget-multisensor tracking: advanced applications*. Artech House, 1990.
- [6] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralised data fusion with distributed particle filters," in *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.
- [7] S. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *9th International Conference on Information Fusion (Fusion'06)*, Florence, Italy, 2006.
- [8] M. Ridley, B. Ucroft, L. Ong, S. Kumar, and S. Sukkarieh, "Decentralised data fusion with Parzen density estimates," in *International Conference on Intelligent Sensors, Sensor Networks and Information Processing 2004 (ISSNIP '04)*, Melbourne, Australia, 2004.
- [9] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer-Verlag, 2001, pp. 247–272.
- [10] N. J. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140(2), pp. 107–113, 1993.
- [11] 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, 2002.
- [12] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer-Verlag, 2001, pp. 3–14.
- [13] M. Coates, "Distributed particle filters for sensor networks," in *Information Processing in Sensor Networks (IPSN 2003)*. Springer, 2004, pp. 99–107.
- [14] X. Sheng, Y. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Fourth International Symposium on Information Processing in Sensor Networks, (IPSN)*, 2005, pp. 181–188.
- [15] A. Ihler, E. Sudderth, W. Freeman, and A. Willsky, "Efficient multi-scale sampling from products of Gaussian mixtures," in *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, 2004.
- [16] S. Challa, M. Palaniswami, and A. Shilton, "Distributed data fusion using support vector machines," in *Proceedings of the Fifth International Conference on Information Fusion, (FUSION)*, vol. 2, 2002, pp. 881–885.
- [17] S. Grime, "Communication in decentralised sensing architectures," Ph.D. dissertation, The University of Oxford, 1992.
- [18] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39.
- [19] M. West, "Approximating posterior distributions by mixtures," *Journal of Royal Statistical Society, Series B*, vol. 55(2), pp. 409–442, 1993.
- [20] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the 17th International Conference on Machine Learning*, p. 277, 2000.
- [21] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications. New York: Wiley, 1991.
- [22] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(5), pp. 564–575, 2003.