Decentralised Particle Filtering for Multiple Target Tracking in Wireless Sensor Networks

Lee-Ling Ong, Tim Bailey, Hugh Durrant-Whyte
ARC Centre of Excellence in Autonomous Systems (CAS)
The University of Sydney, Australia.
Email: s.ong, tbailey, hugh@acfr.usyd.edu.au

Ben Upcroft
Mechanical Engineering Division
The University of Queensland, Australia
Email: ben.upcroft@uq.edu.au

***Abstract*— This paper presents algorithms for consistent joint localisation and tracking of multiple targets in wireless sensor networks under the decentralised data fusion (DDF) paradigm where particle representations of the state posteriors are communicated. This work differs from previous work [1], [2] as more generalised methods have been developed to account for correlated estimation errors that arise due to common past information between two discrete particle sets. The particle sets are converted to continuous distributions for communication and inter-nodal fusion. Common past information is then removed by a division operation of two estimates so that only new information is updated at the node. In previous work, the continuous distribution used was limited to a Gaussian kernel function. This new method is compared to the optimal centralised solution where each node sends all observation information to a central fusion node when received. Results presented include a real-time application of the DDF operation of division on data logged from field trials.**

Keywords: Tracking, estimation, decentralised systems

I. Introduction

In this paper, we consider a situation where a network of sensors perform particle filtering for joint non-linear, non-Gaussian localisation and tracking of multiple targets in the environment. Applied in a decentralised data fusion (DDF) system, the goal is to consistently combine the communicated particle distribution with the local particle distribution of the state posterior of the target under limited communication bandwidth. DDF systems (where there is no central service or facility and communications are made solely to their immediate neighbours) offers the advantages of robustness, scalability and modularity [3].

For a fully decentralised system [4], the estimates communicated, $p(\mathbf{x}_k|\mathbf{Z})$ are comprised of observation information $\mathbf{Z} = \mathbf{z}_1 \dots \mathbf{z}_k$ of the state $\mathbf{x}_k$ at time k obtained from local and remote nodes in the network unlike previous work which communicate observation likelihoods $p(\mathbf{z}_k|\mathbf{x}_k)$ [5], [6]. The main advantage of communicating state posteriors over raw observations or observation likelihoods is bandwidth efficiency. Transmission of likelihoods may result in loss of communicated information or buffering of likelihoods when communication breaks down. However, as observations obtained from the sensors are locally updated with the state estimate at each node, all information contained in the lost messages are implicitly present in future messages received after the communication link is re-opened [4]. Furthermore, instead of communicating and propagating each likelihood message throughout the network to every node, the local observation likelihoods and received state posteriors are combined with the local state posterior, only the combined result is communicated at constant time, reducing communication resource requirements [4].

Most practical applications of distributed and decentralised estimation have been focused on representing features with Gaussians [4], [7]–[10]. However, superior estimation accuracy can be obtained with non-Gaussian estimation when the underlying distribution is not a normal distribution in shape such as a skewed or multi-modal distribution [11], [12] or when there is a need to discriminate and track objects in cluttered environments [13], in which particle filters are an ideal choice [1]. As a result of communicating state posteriors, the estimates are not independent due to correlated or common past observations. Hence, problem faced in fusing non-Gaussian DDF tracks is the removal of common information. In tree structured networks, a division operation is required [3] whereas conservative methods may be applied for networks with loops (redundant communication channels).

The main contributions of this paper are consistent methods based on importance sampling to perform the DDF operation of division. This is demonstrated with particle representations although it is applicable to other non-Gaussian form such as mixture models. The main advantage of these methods compared to that applied in a previous paper [1] is that the transformation to a continuous distribution required for communication and fusion is not restricted to a Gaussian kernel. The accuracy and consistency of the operations based on the Kullback-Leibler (KL) Divergence and entropy is compared to the centralised solution, the closest approximation to the 'true' solution for non-Gaussian representations. The secondary contribution of this paper is the implementation of the DDF operations in real time on data collected from field trails.

II. Related Work

Particle representations have been used for estimation in sensor networks. Information was either communicated to a (central) fusion node [14], [15] or in a decentralised fashion [5], [6], [16]. However, the type of information communicated in the network was limited to either observation

measurements [5], [16]–[18] or state estimates based only on (local) observations made at its corresponding sensor node [6], [15]. In our work, however, the state estimates are based on both local observations and received information from other nodes.

A method to perform fusion on fuse particle representations of these state posteriors is to convert both sets to Gaussian mixtures and applying covariance intersection algorithms in [19] and [20] as demonstrated in our previous work [1], [2]. Although these covariance intersection algorithms aims to ensure conservative fusion of possibly correlated information, they do not guarantee against optimistic estimates.

III. Generalised Decentralised Data Fusion

A decentralised data fusion system (Figure 1) is applied to obtain joint consistent information of target states. Each decentralised node has a sensor which makes observations over which a likelihood is generated and sent to the local filter. The major operations in the node, such as local filtering and node-to-node fusion. The area of primary concern is the node-to-node fusion in which channel filters ensures consistency.

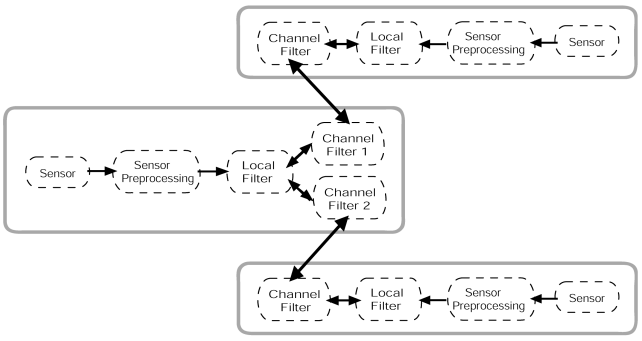


Fig. 1. A decentralised data fusion architecture with channel filters. Local filters update observation likelihoods from local sensors and fuse information received from channel filters. Channel filters maintain a record of common information between two nodes. A separate channel filter is required for every connection, hence there are two channel filters for the node in the middle.

A. Channel Filtering

The fusion between two estimates, where posteriors are communicated is illustrated in [1] amounts to a multiplication and a division operation to remove the common information [21], which is given by:

$$p\left(\mathbf{x}_k|\mathbf{Z}_i \bigcup \mathbf{Z}_j\right) \propto \frac{p\left(\mathbf{x}_k|\mathbf{Z}_i\right) p\left(\mathbf{x}_k|\mathbf{Z}_j\right)}{p\left(\mathbf{x}_k|\mathbf{Z}_i \bigcap \mathbf{Z}_j\right)} \tag{1}$$

For tree-connected networks, channel filters are used to maintain common information about a particular target that a node has in common with another that it is connected to. For each communication link, a different channel filter is required to maintain the common past information between two nodes. As illustrated in Figure 1, when a target state posterior is sent to a connected node, the contents of the local filter would be directly copied to the corresponding channel filter, and the data is then transmitted down the link.

When a message is received at the channel filter, data association is performed with all the local filters to determine which target the data should be associated with. If associated to an existing filter, the following operations are performed:

- Prediction of either the received information in time until it reaches that of the corresponding local filter, or the local filter and channel filters are propagated forward to the corresponding time of the received data.
- Removal of information common between the two nodes from the received message (via a division) before a track-to-track update can be made at the local filter (via a multiplication),
- Update of channel filter so that the received information becomes the new common information.

As a separate channel filter is required for each communication link, the memory resources and scalability of this system is dependent on the number of links a node has. Other drawbacks of channel filtering are that the updates are not robust to incorrect data association and information that may be corrupt, and do not account for the correlation of estimate due to common process noise, which occurs because of the common error source due to the target dynamics [8]. This issue is only prominent when the process noise is large.

IV. Decentralised Data Fusion Operations with Particle Representations

A scalable Gaussian based DDF architecture has been successfully implemented on unmanned ariel vehicles (UAVs) by Nettleton [4], using a Kalman Filter and its Information form. Here, local and communicated information was fused consistently via additive information matrices. However, this methodology does not lend itself to extensions for non-Gaussian distributions including particles.

In addition to the issues of performing the operations of division followed by multiplication or for all generalised representations, the application of particle representations in DDF systems also encounters the following issues which are:

- Particles are weighted Dirac delta functions and hence therefore samples from one particle set do not have the same support on the space as samples from another set. Hence, a naive point wise multiplication or division cannot be applied.
- The communication of particle representations required significantly more bandwidth compared to other representations such as Gaussian mixtures. Bandwidth issues were also addressed in [1] by summarising the particle representation Gaussian kernels. Other candidate representations include Gaussian mixtures [6] and Support Vector Machines [15].

A. Multiplication with Particles

Addressed in previous work [1], node-to-node multiplication can be performed by transforming the particle representations to continuous distributions by placing a Gaussian kernel $K_h(\mathbf{x})$ over each sample [22]:

$$K_h(\mathbf{x}) = h^2 K(\mathbf{x}) \tag{2}$$

where $K(.)$ is the rescaled kernel density and $h > 0$ is the window or scaling parameter. The continuous distribution is then sampled with the second set to obtain the new importance weights so that the weighted sample set represents the multiplication solution.

B. Division with Particles : Proposed Method and Derivation

The division operation in Equation 1 is the key for consistent fusion as the process removes common past information from the received estimate. As with the multiplication of two discrete particle probability distribution functions (PDFs) ($\sum_{i=1}^{N} w_A^{(i)} \delta\left(\mathbf{x}_A^{(i)}\right)$ and $\sum_{i=1}^{N} w_B^{(i)} \delta\left(\mathbf{x}_B^{(i)}\right)$), samples represent densities rather than the underlying function unlike grid representations. Similar to the multiplication operation, the discrete sample sets need to be converted to continuous distributions.

$$\sum_{i=1}^{N} w_A^{(i)} \delta\left(\mathbf{x}_A^{(i)}\right) \rightarrow p_A(\mathbf{x}), \quad \sum_{i=1}^{N} w_B^{(i)} \delta\left(\mathbf{x}_B^{(i)}\right) \rightarrow p_B(\mathbf{x}) \tag{3}$$

An importance sampling approximation is then applied on the continuous distribution. The idea behind Monte Carlo sampling is to draw N i.i.d samples from a distribution $p(\mathbf{x})$, such that the target density is approximated by the following empirical point mass function

$$p(\mathbf{x}) \sim \sum_{i=1}^{N} w_k^{(i)} \delta\left(\mathbf{x}_k^{(i)}\right) \tag{4}$$

where $\delta\left(\mathbf{x}_k^{(i)}\right)$ is the Dirac delta mass located at $\mathbf{x}_k^{(i)}$. Letting $I(f)$ be the expectation of some function f integrable with respect to the PDF $p(\mathbf{x})$ [23]

$$I(f) = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] \triangleq \int f(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{5}$$

the Monte-Carlo approximation of the integral with samples is then

$$I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) \tag{6}$$

where $\mathbf{x}^{(i)} \sim p(\mathbf{x})$ and such that $I_N(f)$ converges to $I(f)$ as $N \rightarrow \infty$.

The true probability distribution $p(\mathbf{x})$ is often hard to sample from, hence, importance sampling is applied.

The idea of importance sampling is to select a proposal distribution $q(\mathbf{x})$ in place of $p(\mathbf{x})$. with the assumption that $q(\mathbf{x})$ encompasses the support space of $p(\mathbf{x})$. The integration problem (Equation 5) is then rewritten as:

$$I(f) = \int f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \int f(\mathbf{x}) w(\mathbf{x}) q(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{7}$$

where $w(\mathbf{x})$, the *importance weights*, is given as

$$w(\mathbf{x}) \triangleq \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{8}$$

A Monte Carlo estimate of $I(f)$ then becomes

$$I_N(f) = \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) w\left(\mathbf{x}^{(i)}\right) \tag{9}$$

For the division operation, the desired probability distribution is

$$p(\mathbf{x}) = \frac{p_A(\mathbf{x})}{p_B(\mathbf{x})} \tag{10}$$

Hence the importance weights in Equation 8 are adjusted to be:

$$w\left(\mathbf{x}^{(i)}\right) = \frac{p_A\left(\mathbf{x}^{(i)}\right)}{p_B\left(\mathbf{x}^{(i)}\right) q\left(\mathbf{x}^{(i)}\right)} \tag{11}$$

which is then normalised so that $\sum_{i=1}^{N} w\left(\mathbf{x}^{(i)}\right) = 1/N$, where N is the number of samples.

C. Division with Particles : Selection of Proposal Distribution

The problem that then arises is the selection of the proposal distribution, $q(\mathbf{x})$. If the proposal distribution selected was a uniform distribution

$$q(\mathbf{x}) = \mathcal{U}(\mathbf{x}) \tag{12}$$

The sample weights $q\left(\mathbf{x}^{(i)}\right)$ are then all equal at a constant value of C. Hence the importance weights are:

$$w\left(\mathbf{x}^{(i)}\right) = \frac{p_A\left(\mathbf{x}^{(i)}\right)}{p_B\left(\mathbf{x}^{(i)}\right) C} \tag{13}$$

Figure 2(b) shows the division operation using 1000 samples drawn from a uniform proposal distribution. While the results are similar to the true distributions, uniform sampling does not scale well in higher dimensions. ($q(\mathbf{x}) = \mathcal{U}_{(-2,8]}(\mathbf{x})$)

Furthermore, there has to be sufficient support over both distributions to perform this operation. Some of the particles may also give negligible contributions to the result, as the space support of those samples is where the distribution has low probability. For example, from the division operation shown in Figure 2(b), the samples lying between $x = 7$ and $x = 8$ provide almost no contribution to the result.

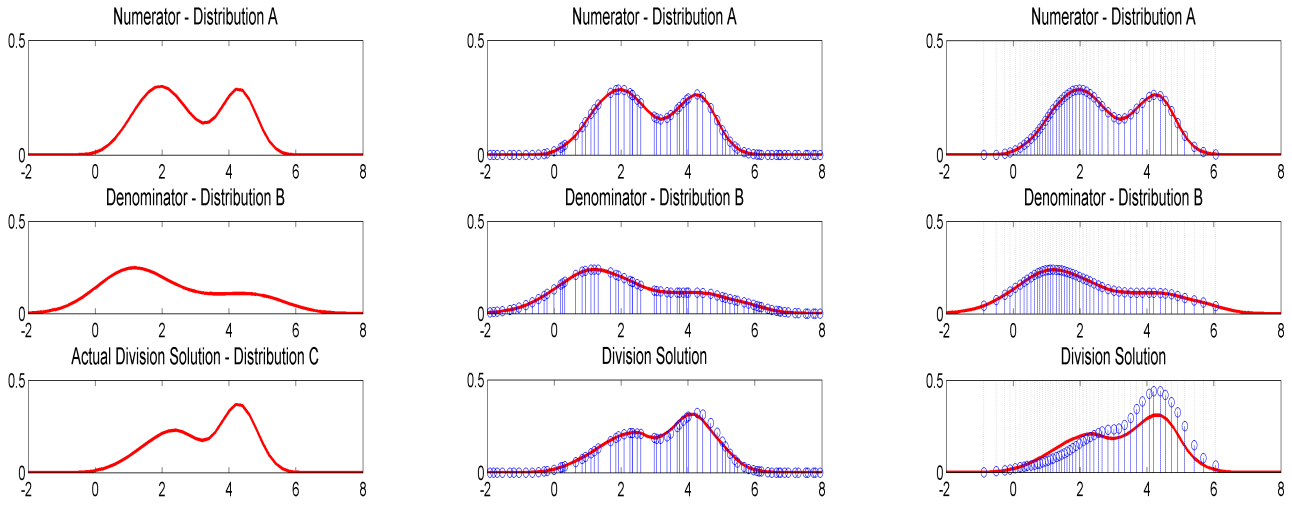
A more scalable solution is to draw samples from another proposal distribution such as a mixture of Gaussians. The main requirement of the proposal distribution is to be able to support the sample space of both the numerator and denominator. It is assumed that node-to-node fusion update occurs on when there is new information received. A received estimate distribution containing the new information would be more tightly clustered. Hence the common information distribution in the channel filter would encompass the received estimate distribution.

Therefore a suitable proposal distribution for the division operation is

$$q(\mathbf{x}) = p_B(\mathbf{x}) \tag{14}$$

The importance weights are then adjusted to be:

$$w\left(\mathbf{x}^{(i)}\right) = \frac{p_A\left(\mathbf{x}^{(i)}\right)}{\left(p_B\left(\mathbf{x}^{(i)}\right)\right)^2} \tag{15}$$



(a) True Distribution.

(b) Uniformly spaced proposal distribution results

(c) Division with weights factored in for samples obtained from a non-uniform distribution.

Fig. 2. The bottom figure of Figure 2(b) shows the result of a division operation using 1000 samples randomly drawn from a uniform distribution while the corresponding figure of Figure 2(c) shows the result when 1000 samples randomly are drawn from from the Distribution B ($p_B(\mathbf{x})$) and weighted according to Equation . Each circle represents the weight of the sample at that point. The particle estimate converted back to a continuous distribution is shown by a solid red line. In both (b) and (c) the results accurately match the true distribution, shown in (a).

Figure 2(c) shows the result of the division with $p_B(\mathbf{x})$ as the proposal distribution and indicate that the division solution is accurate when compared to the actual solution (Figure 2(a)) and the division using samples obtained from a uniformly distributed particle set (Figure 2(b)).

D. Division with Particles: Analysis of Accuracy and Consistency

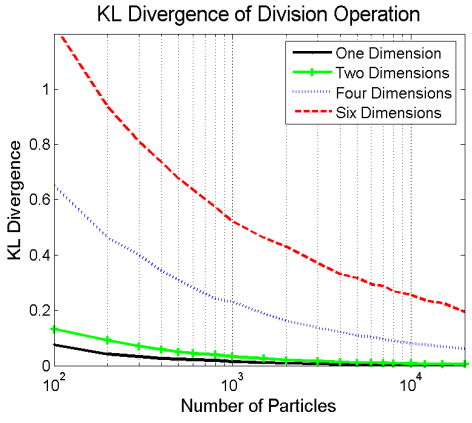


Fig. 3. KL Divergence mean result given the number of particles. As the number of particles increases, the closer the division solution is to the actual one.

With a computational complexity of order N, the division operation is shown to converge to the correct solution as N, the number of particles, increases. Samples are drawn from two known Gaussian mixtures shown as the top two figures in Figure 2.

To analyse the quality of results, two measures are used; the Kullback-Leibler Divergence and the differential entropy. The relative entropy or Kullback-Leibler(KL) divergence between two probability mass functions $p(x)$ and $q(x)$ is [24]:

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \tag{16}$$

The KL Divergence is a measure of shape; it is zero only if the two distributions are identical. However, the KL Divergence is not a true distance between distributions as it is not symmetric and does not satisfy the triangle inequality [24]. However, this allows several approximate distributions to be ranked in terms of their divergence to the "true" distribution

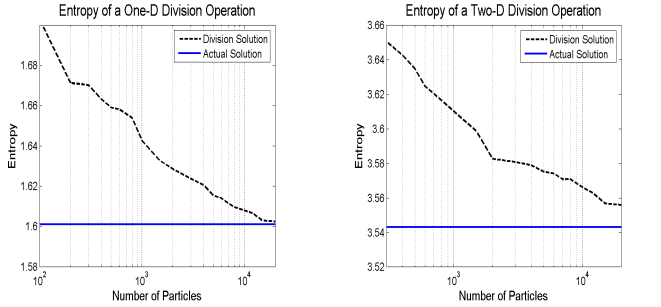
For a probability density, the differential entropy or Shannon entropy is the expected average entropy of the events which is:

$$H(x) = -\mathbf{E}^P\{\log p(x)\} = -\int p(x) \log p(x) \, \mathrm{d}x \tag{17}$$

Entropy is a measure of the compactness of the distribution. If the entropy of the approximate distribution is lower than the true distribution, the estimate is optimistic but if the entropy is higher, nothing definite may be said although a less informative solution may be implied. Entropy is the only reasonable definition of informativeness as shown in [25], [26]. This implies that entropy is an appropriate measure for evaluating probability distributions and should be used relatively. To best measure the accuracy and consistency of the approximations with non-Gaussian distributions, a combination of the KL Divergence and entropy measures are used.

The KL Divergence and entropy computed between the sampled division operation and the actual solution, is shown

in Figures 3, 4 respectively. The KL Divergence results shows that as the number of particles used to perform division increases, the accuracy of the solution increases. The entropy results of the kernel fitted division solution is greater or equal than the actual entropy and converges to the actual entropy at lower dimensions. The higher entropy than the actual result is almost certainly due to the increased variance from adding a kernel over the samples as shown by West [27]. At higher dimensions, the accuracy of the kernel fitting results is a more conservative estimate of the division solution. As the kernel increases the variance of the entire distribution, the resulting approximation tends to be conservative, which is preferable to an optimistic solution.



(a) One-dimensional division sampling mean result given the number of particles.

(b) Two-dimensional division sampling mean result given the number of particles.

Fig. 4. 1-D and 2-D entropy of an approximation of the division operation via sampling vs number of particles.

E. Division with Particles : 4-D DDF Simulation Results

A decentralised bearing-only tracking example was simulated to show the consistency of the division approximation. In this example, a target exhibiting random walk behaviour within the x-y plane was tracked by two stationary sensors providing only bearing information, as shown in Figure 5. The initial state of the feature was $x = 0$, $\dot{x} = 0.5$, $y = 15$ and $\dot{y} = -0.2$. Node 1 was located at $x = 8$, $y = 4$ and Node 2 was located at $x = 5$, $y = 0$.

The feature is modelled based on the Integrated Ornstein-Uhlenbeck process [28] which allows for bounding 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. With a process noise uncertainty of 0.04 in each plane ($q_x = q_y = 0.04$) and a bearing uncertainty of 0.03 radians squared, the prediction and update operations occurred at every second. Two nodes were placed at different positions and perform sequences of bearing measurements. Prediction and update operations occurred at every second. The particle sets were initialised with a range cutoff of 70 m. Each node communicated a summary of its sample set every four seconds.

Thirty Monte-Carlo runs of the simulation were performed. The following scenarios were compared:

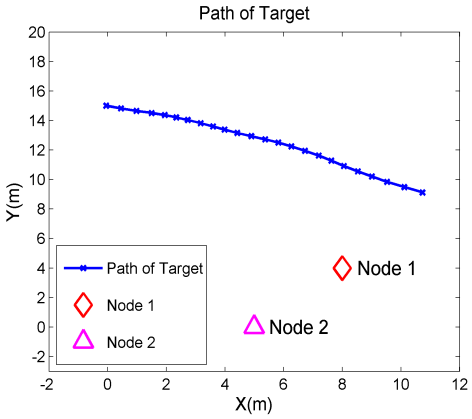


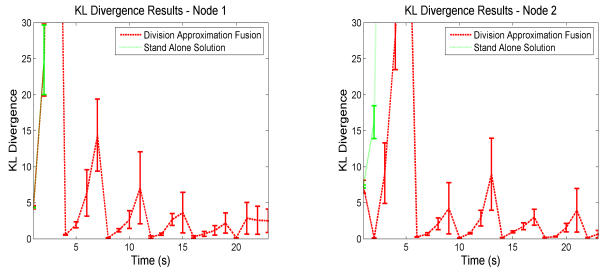
Fig. 5. Path of the feature is shown. Each marker shows the true feature position at a time step. The initial feature position at $x = 0$ and $y = 15$. The stationary sensor nodes are located at $x = 8$, $y = 4$ and $x = 5$, $y = 0$ in the coordinate frame.

- A centralised scenario where each node communicated its likelihood to every other node in the network at every time step. The received and local likelihoods are fused to tracks through local update operations [29].
- A decentralised scenario where each node communicated its feature track (a summary of its sample set) every four seconds. The sample set was converted to kernels as per Equation for fusion using channel filters.
- A stand-alone scenario where no communication of likelihood or estimate occurs.

The performance comparison used for this decentralised simulation is the centralised solution as it provides the closest approximation to the 'true' solution. Here centralised means that each node communicates to every other node in the network at every time step. The samples were converted to Gaussian kernels for fusion.

Hence the KL-Divergence measure [24] is used to show the accuracy of the solution in each decentralised or stand-alone node. The decentralised and stand-alone solutions must contain equal or less information compared to the actual solution, which indicates that information was not double counted using Shannon's entropy measure [26](Equation 17) .

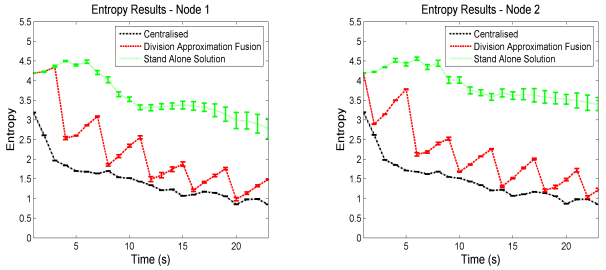
Figure 6 shows the results of the mean and error of KL-Divergence of each particle set over time. The error bars at each time step represent the variance (σ^2) of the Monte-Carlo runs at that time step. Node 1 received Node 2's estimate at the 4th, 8th, 12th, 16th, and 20th second and fusion was performed. Node 1 then communicated its estimate to Node 2 at every 2nd, 6th, 10th, 14th, 18th and 22nd second. At those instances, the KL-Divergence is reduced to nearly zero, indicating that the division solution gives very similar results to the centralised solution. Figure 7 shows the entropy results, which indicate that when fusion is performed, the entropy of the estimate reduces to a value which is equal or slightly more than the centralised solution. This indicates the accuracy of the sampling based division operation, where common past information is removed as no information was double counted.



(a) Node 1 for the stand alone (no Fusion) solution and fusion using a Division Approximation.

(b) Node 2 for the stand alone (no Fusion) solution and fusion using a division approximation.

Fig. 6. Kullback-Leibler Divergence results for both nodes. The nodes performing fusion using a division operation to remove common information exhibit results which are close to or equal to the centralised solution. The nodes not performing fusion result in much larger KL Divergence values which are off the scale of these graphs.



(a) Node 1 for the Stand Alone (no Fusion) solution, Fusion using a Division Approximation and Fusion via a Gaussian mixture covariance intersection.

(b) Node 2 for the Stand Alone (no Fusion) solution, Fusion using a Division Approximation and Fusion via a Gaussian mixture covariance intersection.

Fig. 7. Entropy Results (mean and standard deviation) for both nodes. The standard deviation is shown by the error bars. The division operation is shown remove common information as the entropy is equal or slightly more than the centralised solution when fusion is performed.

Hence, the KL-Divergence and entropy results exhibit a characteristic saw tooth pattern caused by the communication characteristics. The errors bars also increase when fusion is not performed which can be seen in Figure 6. This larger variance is due to the random process noise added during the prediction step as the fusion solution varies from centralised solution also caused by the communication characteristics. When information is shared between nodes at every fourth time step, the solution is near-optimal in comparison with the centralised solution.

V. Real-Time Results for DDF with Particle Representations

Data obtained from ground vehicles (Figure 8(a)) was used to decentralised particle filtering over multiple targets. These ground vehicles were fitted with a firewire colour cameras and INS/GPS subsystems as shown in Figure 8(b). The data sets from two previous ground vehicle logs were used. Each set contains about 15000 video frames logged at 2 Hz with a resolution of 1024 by 768 pixels and logged INS/GPS solutions. Sample video frames are shown in Figure 9. The process and observation models used can be obtained from a previous work [2].

(a) Ground platforms.

(b) Sensors on the ground platforms.

Fig. 8. Ground platform fitted with a firewire colour camera and an INS/GPS subsystem.

Fig. 9. Sampled logged data from the colour cameras showing natural features such as trees and sheds.

The experimental conditions were as follows:

- The platforms represented two ground vehicles supporting one node and one sensor each.
- The sensor data was obtained from two different logs at 2 Hz.
- At intervals of 0.2 Hz, a track was communicated.
- The timescales were at synchronised using NTP and communications was via ethernet.
- The feature extraction algorithms [30], [31] extracted trees and white objects in the environment such as water tanks. Only angular information of the feature position is obtained from the feature extraction results.

The following experimental scenarios were performed:

- A centralised scenario where likelihoods processed at the sensor was communicated to a single node
- A decentralised scenario where each node communicated its feature track
- A stand-alone scenario where no sharing of information occurred to show the estimates of isolated nodes.

When a new target is detected, a filter comprising of 1000 particles is initialised with the first observation as shown in [2]. To detect if tracks were initialised with spurious observations, all filters are maintained as “tentative” tracks for fifteen seconds. After this time, the entropy of the distribution must be below a threshold or the track will be removed. In the decentralised scenario, when an estimate was received and associated to an existing filter, the prediction operation was performed to time-align the data, where the local filter and channel filters were propagated to the corresponding time of the received data. Prior to communication, the particle representations were summarised to Gaussian mixtures of

30 components to reduce bandwidth and as conversion to a continuous distribution is required for a decentralised update.

The centralised solution resulted in more features tracked (due to the entropy threshold heuristic) and more accurately. When track information was shared amongst the nodes, more features are tracked compared to the stand-alone solutions. The number of features was less or equal to the centralised solution. The feature uncertainty is indicated by how tightly clustered a track is. When track information is shared, the solution is less or equally tightly clustered in comparison to the centralised solution. One of the causes of discrepancies between the nodes performing DDF are caused by the communication of one estimate at each communication step. As a result, the uncertainty of a feature track may grow, prior to being selected for communication. This would result in a less informative track communicated and hence more likely to have been pruned. An additional cause of discrepancy was that the conversion of the particle representations to Gaussian mixtures required additional computational resources and time. As there is no sharing of information the maps generated by the nodes acting independently are different from one another. Less features are tracked as less information is available.

VI. Conclusion

New solutions to remove common past information from a received state posterior are introduced so that DDF with channel filtering may be performed consistently with particle filters. Our solution is to transform the particle set to Gaussian mixtures for communication to reduce communication bandwidth and remove known common past information either by a division operation of two estimates so that only new information is updated at the node. The conservativeness and consistency of the division operation and mixture fitting methods are shown using KL Divergence and entropy results. Implementation of the division and multiplication operations for multiple target tracking in real-time is also shown, using data logged from field trials.

Although the channel filter allows common information to be removed accurately, it is limited to a tree connected network. Other drawbacks of channel filtering include not accounting for the correlation of estimate due to common process noise and memory resources requirements as a separate channel filter is required for each communication link, which may be addressed by a conservative fusion update [32], which can also be performed via importance sampling.

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.-L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent methods for decentralised data fusion using particle filters," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, September 2006, pp. 85–91.

[2] L.-L. Ong, B. Upcroft, T. Bailey, M. Ridley, S. Sukkarieh, and H. Durrant-Whyte, "A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, pp. 4539–4544.

[3] S. Grime, "Communication in decentralised sensing architectures," Ph.D. dissertation, The University of Oxford, 1992.

[4] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh, "A robust architecture for decentralised data fusion," in *The 11th International Conference on Advanced Robotics (ICAR'03)*, Coimbra, Portugal, 2003.

[5] M. Coates, "Distributed particle filters for sensor networks," in *Information Processing in Sensor Networks (ISPN 2003)*. Springer, 26-27 April 2004, pp. 99–107.

[6] 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.

[7] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Trans. Automatic Contro,*, vol. 26(2), pp. 802–807, 1981.

[8] K. Chang, R. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33(4), pp. 1271–1276, 1997.

[9] C. Chong, "Distributed architectures for data fusion," in *In Proceedings First International Conference on Multisource-Multisensor Information Fusion*, 1998, pp. 84–91.

[10] L. Y. Pao and M. K. Kalandros, "Algorithms for a class of distributed architecture tracking," in *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997, pp. 1434–1438, distributed Data Fusion Algorithms for Measurements and tracks.

[11] 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.

[12] L. Trailovic and L. Y. Pao, "Variance estimation and ranking of target tracking position errors modeled using gaussian mixture distributions," *Automatica*, vol. 41, no. 8, pp. 1433–1438, August 2005.

[13] J. S. Williams, "Gaussian mixture reduction for tracking multiple maneuvering targets in clutter," Ph.D. dissertation, Deparment of the Air Force, Air University, 2003.

[14] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms and architectures for distributed particle filters," *IEEE Transactions on Signal Processing*, vol. 53(7), pp. 2442–2450, June 2005.

[15] S. Challa, M. Palaniswami, and A. Shilton, "Distributed data fusion system using support vector machines," in *Proceedings IEEE Conference on Information Fusion (FUSION2005)*, Annapolis , USA, 2002.

[16] 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.

[17] A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," *Information Processing in Sensor Networks, Berkeley, California, USA*, 2004.

[18] M. Rosencrantz, G. Gordon, and S. Thrun, "Locating moving entities in dynamic indoor environments with teams of mobile robots," in *Proceedings of Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003.

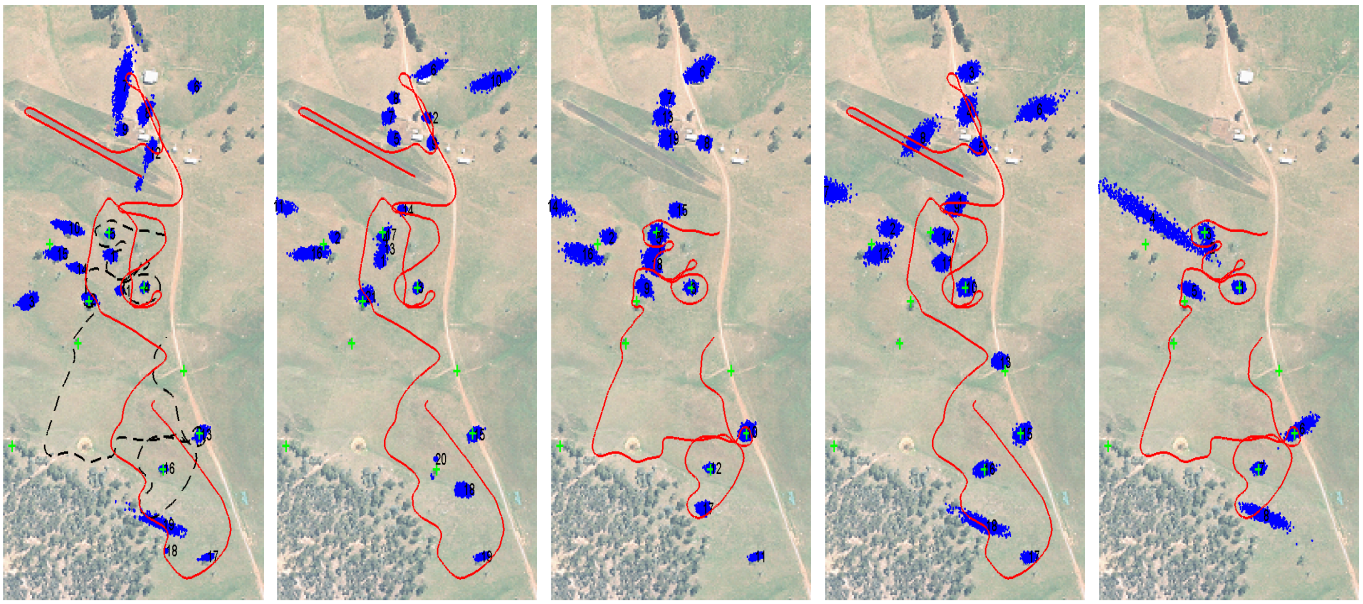
[19] S. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *9th International Conference on Information Fusion (FUSION '06)*, Florence, Italy, 2006.

[20] B. Upcroft, M. Ridley, L. Ong, B. Douillard, T. Kaupp, S. Kumar, T. Bailey, F. Ramos, A. Makarenko, A. Brooks, S. Sukkarieh, and H. Durrant-Whyte, "Multi-level state estimation in outdoor decentralised sensor network," in *Proceedings of the International Symposium on Experimental Robotics*, Rio de Janiero, Brazil, June 2006.

[21] Y. Bar-Shalom, *Multitarget-multisensor tracking: advanced appplications*. Artech House, 1990.

[22] 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.

[23] 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.



(a) Centralised Solution (b) Node 1 performing DDF (c) Node 2 performing DDF (d) Node 1 without DDF (e) Node 2 without DDF

Fig. 10. Snapshots of the final results using particle distributions. The solid line represents the vehicle path. The dashed line in the figure showing the centralised solution represents the second vehicle path. The surveyed differential GPS location of features are marked as crosses. However, other features such as sheds and water tanks are also tracked.

[24] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications. New York: Wiley, 1991.

[25] D. E. Catlin, *Estimation, Control and the Discrete Kalman Filter*, L. S. F. John, J.E. Marsden, Ed. Berlin: Springer-Verlag, 1989, vol. 71.

[26] C. Shannon, "A mathematical theory of communication," *Bell Systems Techical Journal*, vol. 2,7, pp. 379–423,623–656, July and October 1948.

[27] M. West, "Approximating posterior distributions by mixtures," *Journal of Royal Statistical Society, Series B*, vol. 55(2), pp. 409–442, 1993.

[28] L. Stone, C. Barlow, and T. Corwin, *Bayesian Multiple Target Tracking*. Artech House, 1999.

[29] 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.

[30] S. Kumar, F. T. Ramos, B. Douillard, M. Ridley, and H. Durrant-Whyte, "A novel visual perception framework," in *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, (ICARCV '06)*, Singapore, December 2006, pp. 824–829.

[31] S. Kumar, F. T. Ramos, B. Upcroft, and H. Durrant-Whyte, "A statistical framework for natural feature representation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, Edmonton, Canada, 02 Aug. - 06 Aug 2005.

[32] S. Julier, T. Bailey, and J. Uhlmann, "Using exponential mixture densities for suboptimal distributed data fusion," in *The 1st IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW '07)*, Cambridge, UK, 13 15 September 2006.