DISTRIBUTED PARTICLE FILTERS FOR WIRELESS SENSOR NETWORK TARGET TRACKING

Xiaohong Sheng and Yu-Hen Hu

Department of Electrical and Computer Engineering
University of Wisconsin - Madison, WI 53706, USA
sheng@ece.wisc.edu, hu@engr.wisc.edu

ABSTRACT

We propose two distributed particle filters to estimate and track the moving targets in a wireless sensor network. The observations by the sensors are divided into a set of disjoint uncorrelated cliques. The first distributed algorithm runs the local particle filters sequentially at each clique. The second distributed algorithm runs the local particle filters in parallel to obtain the local sufficient statistics, and then send these statistics to a centralized location through multi-hops to obtain the final estimates. The two distributed algorithms are both convergence almost surely. In addition, we proposed to use local Gaussian mixture model (GMM) to approximate the posteriori distribution obtained from the local particle filter. By propagating the GMM parameters rather than belief, we achieve significant bandwidth and power consumption reduction. Very promising simulation results are reported as well.

1. INTRODUCTION

In [1], we proposed a Centralized Particle Filter(CPF) to sequentially localize and track the targets in a wireless sensor network. The algorithm is to partition the sensor field into fixed-size regions and run a centralized particle filter algorithm within the region. While it is simple to implement and is capable of delivering robust performance for target tracking, CPF requires high communication because the observation of each node has to be delivered to a region center.

In this paper, we present two novel Distributive Particle Filter (DPF) algorithms to facilitate efficient implementation of a particle filter based sequential target tracking system over a wireless sensor network. Our approach is to distribute the computation burden and communication burden over the entire sensor network. Specifically, we divide the sensor field into small uncorrelated "cliques" of sensor clusters. Each clique of sensors can communicate each other with local broadcasting via wireless channel. Hence the communication cost within each clique is relatively low. Two versions of distributed particle filtering algorithms, denoted by DPF-I and DPF-II are proposed: With DPF-I, the importance weights are updated from one clique of sensors to the next. Each clique's estimate is built upon all preceding cliques' partial observations and estimates. These partial estimates then will be forwarded via wireless communication channel to the next clique with one hop communication. With DPF-II, each clique updates its own importance weight estimate in parallel based only on their local observations. These partial estimates are then transmitted over the entire active sensor region through multi-hops to update the final posterior distribution.

To facilitate these updates, the belief estimates must be communicated among sensor cliques via wireless communication channel. These requirements would defeat the purpose of distributive particle filter processing since it will require transmitting large amount of information. To reduce this communication burden, we propose to approximate the belief estimates with a low dimension Gaussian mixture model (GMM). Instead of transmitting raw estimates of particles, we transmit the mixture Gaussian model parameters. This approximation scheme significantly reduces the demand on the communication bandwidth.

Our distributed particle filters are quite different from the few existing distributed particle filter approaches. In particular, the work reported in [2] sought to update the complete particle filter on each individual sensor nodes. The algorithm requires the independency among all sensor observations. Further, very complicated learning procedures must be performed before running their algorithm. Ours algorithms, on the other hand, seek to derive the complete particle filter estimates over the final sensor node clique. The designation of the final clique may be dynamically changed based on the predicted target trajectories. Further, our distributed algorithm does not require learning procedure and the sensor observations are not necessary to be independent between each other.

2. DISTRIBUTED PARTICLE FILTER

2.1. Notation

Let $X = \{x_t, t \in N\} \in \Re^{n_x}$ be a stochastic process defined over a probability space $(\Omega, \mathcal{F}, P)$, where n_x is the dimension of X. We also assume that X is a Markov process such that $P(x_{t+1} \in A|x_{1:t}) = P(x_{t+1} \in A|x_t)$. The transition kernel of the Markov chain is defined as: $K_t(x, A) = P(x_{t+1} \in A|x_t = x)$

The observation model is: $\mathbf{y}_t = \mathbf{h}_t(x_t, w_t)$, where the observation noise w_t is assumed to be independent with the state vector x_t. Further, we assume that the observation $\mathbf{y}_t$ can be divided into a set of disjoint uncorrelated cliques, i.e. $\mathbf{y}_t = [\mathbf{y}_{t,1}, \mathbf{y}_{t,2}, ..., \mathbf{y}_{t,M}]^T$.

We denote $\hat{p}_t = P(x_t|\mathbf{y}_{0:t-1})$, $\pi_t = P(x_t|\mathbf{y}_{0:t})$, $\pi_{t,k} = P(x_t|\mathbf{y}_{0:t-1}, \mathbf{y}_{t,1:k})$. Similarly, we denote $\hat{p}_t^n$ and π_t^n as the prior and posterior probability estimated by centralized particle filter and $\pi_{t,k}^n$ and $\hat{p}_{t,k}^n$ as the distributed prior and posterior probability respectively.

2.2. Distributed particle filter algorithms

In general, most of particle filters require three steps to estimate the posterior distribution, i.e., initialization, prediction and update [3] [4].

During initialization, n random particles are uniformly drawn from the initial prior distribution π_0. By strong law of large number, $\lim_{n\to\infty} \pi_0^n \to \pi_0$ a.s.

In the prediction step, the new positions of the particles are computed based on the transition kernel. In the update step, the weights are calculated for each particle according to the new received observations. In our proposed distributed particle filter, the update step is performed by local particle filters that are composed of a set of uncorrelated cliques.

The information sensed by the nodes in sensor network is governed by the events in the sensor field. Each signal received by the sensor located at (a_x, a_y) corresponds to a space-time signal with spatial bandwidth B_{a_x}, B_{a_y} and temporal bandwidth B. The coherence distance $D_{a_x} = \frac{1}{B_{a_x}}$ and $D_{a_y} = \frac{1}{B_{a_y}}$ denotes the spatial dimension over which the signals are strongly correlated [5]. $D_{a_x} \times D_{a_y}$ determines spatial coherence region, i.e., clique. The signals are approximately uncorrelated in distinct clique, i.e. $\pi_t \approx \prod_{k=1}^{M} \pi_{t,k}$, where M is the number of uncorrelated cliques.

For signal produced by point source such as targets, the spatial signal bandwidth is determined by the temporal bandwidth via the speed of signal propagation. For isotropic spatial propagated signal, $s(a_x, a_y, t) = s(0, 0, t - \frac{\sqrt{{a_x}^2 + {a_y}^2}}{v})$, where v is the speed of propagation. The spatial bandwidth and coherence distance in the radial coherence dimension are:

$$B_r = \frac{B}{v};\ D_r = \frac{1}{B_r} = \frac{v}{B} \tag{1}$$

where $r = \sqrt{{a_x}^2 + {a_y}^2}$

Coherence distance is much smaller for sound propagation than that for magnetic wave propagation. Thus, when we apply acoustic signal for target localization and tracking, the active sensor region can be divided into several small uncorrelated cliques. This feature makes it possible to run the distributed particle filter efficiently.

2.2.1. *Distributed Particle Filter Algorithm I: DPF-I*

The predicted particles distribute as: $x_{t,0}^i \sim K_t\left(x_{t-1,M}^i, \cdot\right)$, where $K_t\left(x_{t-1,M}^i, \cdot\right)$ is the transition kernel, $x_{t-1,M}^i$ and $x_{t,0}^i$ are respectively, the current and predicted particle positions.

Distributed algorithm 1 updates the posterior distribution sequentially by a sequence of local particle filters, i.e. For $k = 1$ to M,

Denote $\bar{x}_{t,k}^i = x_{t,k-1}^i$,

Compute

$$w_{t,k}^i = g_{t,k}\left(\bar{x}_{t,k}^i\right) / \sum_{j=1}^{n} g_{t,k}\left(\bar{x}_{t,k}^j\right) \tag{2}$$

where $g_{t,k}(\bar{x}_{t,k}^i) = P(\mathbf{w} = \mathbf{y}_{t,k} - \mathbf{h}_{t,k}(\bar{x}_{t,k}^i))$.

Resample particles by replacing $\bar{x}_{t,k}^i$ with a number of offspring $n_{t,k}^i$ according to a multinomial distribution of parameters $w_{t,k}^i$ and $\sum_{i=1}^{n} n_{t,k}^i = n$. Denote the resampled particles as $x_{t,k}^i$, the approximated posterior distribution before and after resampling are:

$$\begin{aligned} \bar{\pi}_{t,k}^n &= \sum_{i=1}^{n} w_{t,k}^i \delta_{\{\bar{x}_t^i\}} \\ \pi_{t,k}^n &= \frac{1}{n} \sum_{i=1}^{n} n_{t,k}^i \delta_{\{\bar{x}_t^i\}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\{x_{t,k}^i\}} \end{aligned}$$

Theoretically, resampling procedure is not necessary at each local DPF. However, transmitting these particles as well as weights of these particles among the local DPFs needs high communication. To reduce the communication, the posterior distribution(particles and weights) is represented by a low dimensional GMM. Hence, only GMM parameters need to be transmitted among the local DPFs. Resampling procedure is performed to facilitate the estimation of GMM parameters.

We have proved that $\lim_{n\to\infty} \pi_{t,M}^n \to \pi_t$ almost surely and the convergence rate is $\frac{M}{\sqrt{n}}$. Due to space limitation, these proofs are omitted here.

2.2.2. *Distributed Particle Filter Algorithm II: DPF-II*

In stead of updating the posterior distribution sequentially using a sequence of local particle filters, the DPF-II runs the local particle filters in parallel to obtain the local sufficient statistics and estimate the final posterior distribution based on these local sufficient statistics.

At every clique, parallel compute

$$w_{t,k}^i = \frac{g_{t,k}\left(\bar{x}_t^i\right)}{\sum_{j=1}^{n} g_{t,k}\left(\bar{x}_t^j\right)} \tag{3}$$

where $\bar{x}_t^i \sim K_{t-1}(x_{t-1}^i, .)$

Note that $g_{t,1:M}\left(\mathbf{x}_t\right) = \prod_{k=1}^{M} g_{t,k}\left(\mathbf{x}_t\right)$ when cliques are uncorrelated between each other, the update weight at time t is:

$$w_t^i = \frac{g_{t,1:M}\left(\bar{x}_t^i\right)}{\sum_{j=1}^n g_{t,1:M}\left(\bar{x}_t^j\right)} = \frac{g_{t,1}\left(\bar{x}_t^i\right)\prod_{k=2}^M g_{t,k}\left(\bar{x}_t^i\right)}{\sum_{l=1}^n g_{t,1}\left(\bar{x}_t^l\right)\sum_{j=1}^n \frac{\prod_{k=1}^M g_{t,k}\left(\bar{x}_t^j\right)}{\sum_{l=1}^n g_{t,1}\left(\bar{x}_t^l\right)}}$$
$$= \frac{w_{t,1}^i \prod_{k=2}^M g_{t,k}\left(\bar{x}_t^i\right)}{\sum_{j=1}^n w_{t,1}^j \prod_{k=2}^M g_{t,k}\left(\bar{x}_t^j\right)} = \dots = \frac{\prod_{k=1}^M w_{t,k}^i}{\sum_{j=1}^n \prod_{k=1}^M w_{t,k}^j} \quad (4)$$

Eq. (4) shows that DPF-II yields identical particle filter estimates as the centralized implementation. The convergence rate for DPF-II is $1/\sqrt{n}$.

3. GAUSSIAN MIXTURE APPROXIMATION

3.1. Gaussian Mixture Approximation

Both distributed DPF-I and DPF-II require belief propagation, which may need to transmit more bits than raw observation data. In this paper, we propose to approximate the distribution by Gaussian Mixture Model (GMM).

$$\pi_{t,k}^n \simeq \hat{\pi}_{t,k}^n = \sum_{m=1}^{c} \hat{\lambda}_{t,k}^m \mathcal{N}(\hat{\mu}_{t,k}^m, \hat{\sigma}_{t,k}^m) \quad (5)$$

where c is the number of mixtures. Thus, the belief can be propagated through the transmission of the parameters of GMM $\left(\hat{\lambda}_{t,k}^m, \hat{\mu}_{t,k}^m, \hat{\sigma}_{t,k}^m\right)$ rather than the particles and weights. The communication burden is dramatically reduced.

The parameters of GMM are estimated using EM algorithm [7]. Using Lagrange multiplier, one may find:

Expectation step:

$$\hat{\lambda}_{t,k}^m = \frac{1}{n}\sum_{i=1}^{n} \hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right) \quad (6)$$

$$\hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right) = \frac{\mathcal{N}\left(x_{t,k}^i, \hat{\mu}_{t,k}^m, \hat{\sigma}_{t,k}^m\right)\hat{\lambda}_{t,k}^m}{\sum_{l=1}^c \mathcal{N}\left(x_{t,k}^i, \hat{\mu}_{t,k}^l, \hat{\sigma}_{t,k}^l\right)\hat{\lambda}_{t,k}^l}$$

Maximization step:

$$\hat{\mu}_{t,k}^m = \frac{\sum_{i=1}^n x_{t,k}^i \hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right)}{\sum_{i=1}^n \hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right)}$$

$$\hat{\sigma}_{t,k}^{m2} = \frac{\sum_{i=1}^n \left(x_{t,k}^i - \hat{\mu}_{t,k}^m\right)^2 \hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right)}{\sum_{i=1}^n \hat{\lambda}_{t,k}\left(m|x_{t,k}^i\right)} \quad (7)$$

By incorporating this GMM approximation, the DPF-I and DPF-II algorithm can be modified accordingly.

Note that for DPF-II, each clique center runs the algorithm in parallel and passes the local GMM parameters to the region center through multi-hop communication (normally, we can dynamically pick one of the clique center as the region center). The updated posterior distribution through each hop is the normalized multiplication of local GMMs, which is c^2 mixtures of Gaussian. We further use the k-mean algorithm [8] to cluster the c^2 mixtures of Gaussian into a c-mean mixture of Gaussian and forward it to the next hop. The detailed implementation is omitted due to the space limitation.

The number of c can be selected according to the number of targets. Since we can merge targets together when they are close to each other, the number of distinct targets is small.

3.2. Distributed communication in sensor network

To reduce the communication cost, the nodes within the clique broadcast so that the clique center can receive the observation of its member nodes. The power for broadcasting can be low since the size of the clique is small (normally it is less than 30 meters for the acoustic signal). Each node will be set a backoff time before broadcasting to avoid interfering between each other. The backoff time we designed is inverse proportional to its SNR.

The clique center will pass GMM parameters using pear to pear communication. The communication cost in bits per time step is: $\sum \mathcal{G}H_k$, where $\mathcal{G}$ is the number of bits required to represent the parameters of local GMM, H_k is the number of communication hops required to pass the local GMM parameters. $H_k = 1$ for DPF-I. The maximum of H_k is M for DPF-II.

3.3. Comparison of the two algorithms

As stated above, these two DPF algorithms convergence almost surely to the true posterior probability. The convergence rate of DPF-II is faster than that of DPF-I. In fact, from Eq.(4), we know that the distributed particle filter with DPF-II has no loss compared with the centralized particle filter. Therefore, to get same performance, DPF-I requires more particles than DPF-II. However, DPF-II requires higher communication than DPF-I.

4. SIMULATION ON TWO TARGETS TRACKING IN WIRELESS SENSOR NETWORK

Consider the tracking of moving targets over a 2-dimensional sensor field of the size $100 \times 100m^2$. The Markov state transition model is described as follows:

$$\begin{aligned} \mathbf{a}_l(t) &= \mathbf{v}(t) \\ \mathbf{u}_l(t) &= \mathbf{u}_l(t-1) + \mathbf{a}_l(t)T \\ \boldsymbol{\rho}_l(t) &= \boldsymbol{\rho}_l(t-1) + \mathbf{u}_l(t-1)T + \frac{1}{2}\mathbf{a}_l(t)T^2 \end{aligned}$$

where $\boldsymbol{\rho}_l(t)$ stands for the location of source l, $\mathbf{u}_l(t)$ is the velocity of source l and $\mathbf{a}_l(t)$ is the acceleration of the source l. T is the time interval and $\mathbf{v}(t)$ is assumed to

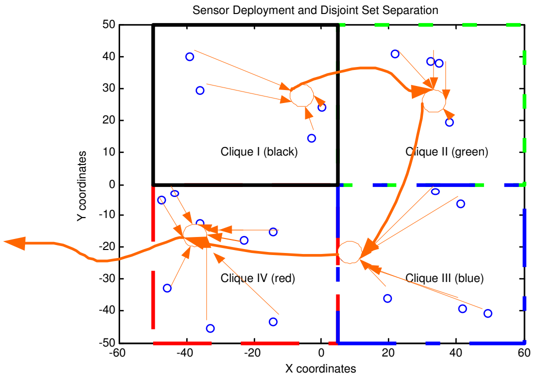

Fig. 1. Sensor Placement for Target Localization Estimation Simulation (Big node is the clique center)

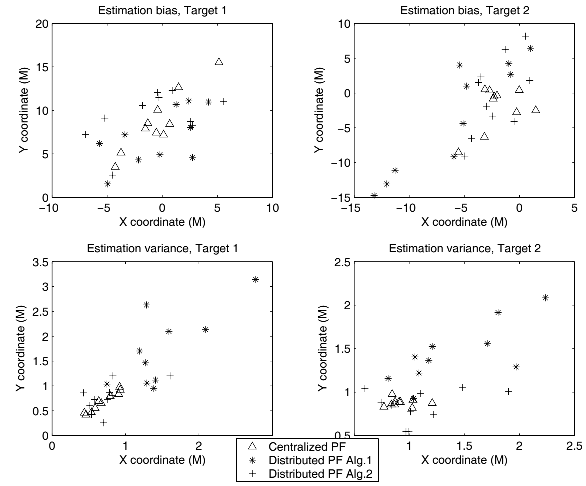

Fig. 2. Estimation Bias and Variance using Centralized and Distributed Particle Filter estimation.

be uniformly distributed on $[-A_{max} \;\; A_{max}]$. A_{max} is the maximum acceleration rate.

In [6], we proposed the intensity (energy) based estimation model for source localization that has been validated through real world experiment:

$$y_i(t) = \gamma_i \sum_{l=1}^{L} \frac{s_l(t)}{\| \boldsymbol{\rho}_l(t) - \mathbf{r}_i \|^2} + \varepsilon_i(t) \qquad (8)$$

where L is the number of targets (assumed to be known), $y_i(t)$ is the acoustic energy received by the i^{th} sensor at time t. $\varepsilon_i(t)$ is a perturbation term that summarizes the net effects of background additive noise and the parameter modeling error. γ_i and $\mathbf{r}_i$ are the gain factor and location of the i^{th} sensor respectively, $s_l(t)$ is the energy emitted by the l^{th} source during the t^{th} time interval. We have analyzed [6] the probability distribution of $\varepsilon_i(t)$ and concluded that it can be modeled well with an $i.i.d.$ Gaussian random variable when the time window T for averaging the energy is sufficiently large. The mean and variance of each $\varepsilon_i(t)$ can be empirically estimated from constant false alarm (CFAR) detector. The sensors are randomly deployed as Fig.1. The whole sensor field is divided into four disjoint sensor cliques. The source energy for target 1 measured at 1 meter distance is set as $s_1 = 10000$. The source energy for target 2 is set as $s_2 = 1.2s_1$. The background noise level is set as $\sigma_i = 3$ for all sensors in the sensor field. The number of particles is chosen to be 500. 500 repeated trials are simulated for each sequential running point. Estimation bias and variance are shown in Fig. 2. Results show that performance of the distributed of particle filter is almost the same as the centralized algorithm. As our analysis, DPF-II has better performance than DPF-I.

5. CONCLUSION

Two distributed particle filters are proposed to localize and track the targets in wireless sensor network. The algorithms are run distributively at local clique to obtain the local sufficient statistics. By approximating the local sufficient statistic (belief) with GMM, the communication burden is dramatically reduced while the performance is maintained almost the same as the centralized particle filter. The algorithm can further be improved by grouping the targets and reducing the number of particles through a deterministic kernel placement.

6. REFERENCES

[1] X. Sheng and Y.H. Hu, “Sequential acoustic energy based source localization using particle filter in a distributed sensor network”, ICASSP2004, III-972-975

[2] Mark Coates, “Distributed particle filters for sensor networks” Information Processing in Sensor Networks, IPSN2003, Springer, pp:99-107, 2004

[3] A.Doucet, N. de Freitas, and N. Gordon, editors. Sequential Monte Carlo Methods in Practice. Springer-Verlag, 2001

[4] Petar M. Djuric, etal, “Particle Filtering”, IEEE Signal Processing Magazine, v.20, Sep. 2003, pp.19-38

[5] A. Sayeed, ”A Statistical Signal Modeling Framework for Sensor Networks” , Technical Report, 2004

[6] X. Sheng and Y.H. Hu, “Energy based source localization”, Information Processing in Sensor Networks, IPSN2003, Springer, pp:285-300, 2003

[7] Moon, T.K.; “The expectation-maximization algorithm”, Signal Processing Magazine, IEEE , Volume: 13 , Issue: 6 , Nov. 1996 Pp:47-60

[8] Simon Haykin,“Neural Networks: A Comprehensive Foundation”, Prentice Hall, New Jersey, second edition, 1999