Signal Processing 95 (2014) 149–160

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Belief consensus algorithms for fast distributed target tracking in wireless sensor networks

Vladimir Savic [a,b,*], Henk Wymeersch [c], Santiago Zazo [a]

[a] *Signal Processing Applications Group, Universidad Politecnica de Madrid, Spain*
[b] *Department of Electrical Engineering (ISY), Linköping University, Sweden*
[c] *Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden*

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
28 July 2013
Accepted 3 September 2013
Available online 13 September 2013

Keywords:
Belief consensus
Belief propagation
Distributed target tracking
Particle filtering
Wireless sensor networks

ABSTRACT

In distributed target tracking for wireless sensor networks, agreement on the target state can be achieved by the construction and maintenance of a communication path, in order to exchange information regarding local likelihood functions. Such an approach lacks robustness to failures and is not easily applicable to ad-hoc networks. To address this, several methods have been proposed that allow agreement on the global likelihood through fully distributed belief consensus (BC) algorithms, operating on local likelihoods in distributed particle filtering (DPF). However, a unified comparison of the convergence speed and communication cost has not been performed. In this paper, we provide such a comparison and propose a novel BC algorithm based on belief propagation (BP). According to our study, DPF based on metropolis belief consensus (MBC) is the fastest in loopy graphs, while DPF based on BP consensus is the fastest in tree graphs. Moreover, we found that BC-based DPF methods have lower communication overhead than data flooding when the network is sufficiently sparse.

1. Introduction

Distributed tracking in wireless sensor networks (WSNs) [1] is an important task for many applications in which central unit is not available. For example, in emergency situations, such as a fire, a nuclear disaster, or a mine collapse, a WSN can be used to detect these phenomena. Once a phenomenon is detected (e.g., increased temperature or radioactivity), the sensors start to sense their neighborhood and cooperatively track people and assets. As sensors are low-cost devices that may not survive during tracking, it is important to track in a manner that is fully robust to sensors failures, and in such a way that every sensor has the same belief of the target location. Then, the rescue team can access the estimates, even if just one sensor survives. As another potential application, sensor nodes can also serve as actuators, which perform a specific action (e.g., move towards the target) as a function of estimated target's position. In this case, to ensure compatible actions, a unified view of the target's position is crucial.

The traditional approach to target tracking is based on Kalman filtering (KF) [2]. However, due to nonlinear relationships and possible non-Gaussian uncertainties, a particle filter (PF) is preferred [3] in many scenarios. Therefore, the focus of this paper will be on PF-based distributed tracking. Many PF-based methods are based on the construction and maintenance of a communication path, such as a spanning tree or a Hamiltonian cycle. For example, in [4], low-power sensors pass the parameters of likelihood functions to high-power sensors, which are responsible to manage the low-power nodes. In [5], a set of uncorrelated sensor cliques is constructed, in which slave nodes have to transmit

* Corresponding author at: Department of Electrical Engineering (ISY), Linköping University, Sweden.
E-mail addresses: vladimir.savic@liu.se (V. Savic), henkw@chalmers.se (H. Wymeersch), santiago@gaps.ssr.upm.es (S. Zazo).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.09.005

Gaussian mixture parameters to the master node of the clique. The master node performs the tracking, and forward estimates to another clique. In [6], a Markov-chain distributed PF is proposed, which does not route the information through the graph during tracking. However, it requires that each node knows the total number of communication links and the number of communication links between each pair of nodes, which can be obtained only by aggregating the data before tracking. In [7], the authors propose an incremental approach, in which the parameters of the likelihood are communicated from sensor to sensor in order to approximate the posterior of interest. Finally, there is also a different class of methods [8,9] that maintain disjoint sets of particles at different nodes, and propagate them towards the predicted target position. These type of methods, also known as *leader-agent algorithms* (see [1] for an overview), lack robustness to failures, cause excessive delays due to the sequential estimation, and do not provide the estimates at each sensor without additional post-processing routing phase.

These problems can be solved if each node broadcasts observations until all the nodes have complete set of observations. Then, each node (acting like a fusion center) performs the tracking. This method, known as *data flooding* and used in non-centralized PF (NCPF) [10], is not scalable, but can be competitive in some scenarios. Other solutions consider distributed particle filtering (DPF) methods based on consensus algorithms [11–17]. In [11], the global posterior distribution is approximated with a Gaussian mixture, and consensus is applied over the local parameters to compute the global parameters. Similarly, [12,13] use a Gaussian approximation instead of a Gaussian mixture, and [14] can use any distribution that belongs to an exponential family. Randomized gossip consensus was used in [15] for distributed target tracking. The main problem with these approaches is that the global likelihood function is represented in the same parametric form as local likelihood functions, which is questionable in certain scenarios. In [16,17], consensus is applied instead to the weights in the DPF, so that any likelihood can be represented. However, an issue that arises with these DPF approaches is that consensus can be slow. In a setting where the target moves, only a finite time is available to perform consensus [18], so the fastest possible method should be employed. A recent and detailed overview of DPF algorithms can be found in [1], but it does not analyze the effect of different consensus techniques on convergence.

In this paper, we compare five algorithms for target tracking using distributed particle filtering (DPF) based on belief consensus (BC):

1. standard belief consensus (SBC) [17];
2. randomized gossip (RG) [16];
3. broadcast gossip (BG) [19];
4. Metropolis belief consensus (MBC) [14]; and
5. one novel algorithm based on belief propagation (BP), which we earlier proposed in [20].

To the best of our knowledge, this is the first study where these methods are compared in a common setting. According to our simulation study, DPF-MBC is the fastest in loopy graphs, while DPF-BP is the fastest in tree graphs (typical for tunnel-like environments). Moreover, we found that BC-based DPF methods have lower communication overhead than data flooding only in sparse networks.

The rest of this paper is organized as follows. In Section 2, we review centralized target tracking. In Section 3, we describe five BC algorithms for PF-based distributed target tracking, including the novel based on BP. Simulation results are shown in Section 4. Finally, Section 5 provides our conclusions and suggestions for future work.

2. Overview of centralized target tracking

We assume that there is a number of static sensor nodes with known positions and one moving target (e.g., a person or vehicle) in some surveillance area. The target may be passive or not willing to reveal its state, but the sensors are assumed to periodically make observations that depend on their relative position to the target. The goal of the WSN is to track the state (e.g., position and velocity) of the target. In this section, we describe a centralized approach to solve this problem, in which all the observations are collected by a sensor that acts as a fusion center.

2.1. System model

The scenario under consideration is illustrated in Fig. 1. There are N_s static sensors with known two-dimensional (2D) positions, $\mathbf{l}_n$ $(n = 1, 2, \ldots, N_s)$ and one mobile target with an unknown state $\mathbf{x}_t$ at time t. The goal of the WSN is to estimate $\mathbf{x}_t$ at each (discrete) time t. We use the following state-space model:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) \quad (1)$$

$$y_{n,t} = g_n(\mathbf{x}_t, v_{n,t}), \quad (2)$$

where $\mathbf{u}_t$ is process noise, $y_{n,t}$ is local observation of sensor n at time t, and $v_{n,t}$ is its observation noise. We denote the aggregation of all observations at time t by $\mathbf{y}_t$. The process noise $\mathbf{u}_t$ can be non-Gaussian, but since it is usually hard to measure [2,21], we can assume a Gaussian approximation with sufficiently large variance, which is a common choice.

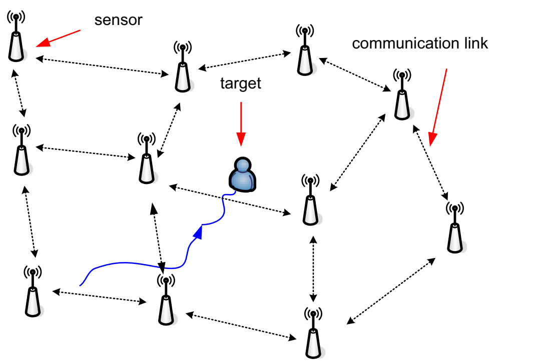

Fig. 1. Illustration of target tracking in a WSN. The goal of the WSN is to track the position and velocity of the target.