CONSENSUS-BASED DISTRIBUTED ESTIMATION IN CAMERA NETWORKS

Ahmed T. Kamal, Jay A. Farrell and Amit K. Roy-Chowdhury

University of California, Riverside.

ABSTRACT

Distributed algorithms in the sensors networks community usually require each sensor to have its own measurement. In practice, this constraint can not always be met. For example, in a camera network, all cameras might not observe a particular target as cameras are directional sensors and have a limited field-of-view (FOV). Moreover, different sensors might provide different quality measures related to different elements of the measurement vector depending on various factors as directionality, occlusion etc. This requires the designing of a new type of distributed algorithm that considers the quality and/or absence of measurements. In this paper, we present a distributed algorithm to compute the maximum likelihood estimate of the state of a target viewed by the network of cameras, taking into account the above-mentioned factors. We provide step-by-step derivation along with theoretical guarantee of optimality and convergence of the method. Experimental results are provided to show the performance of the proposed algorithm.

Index Terms— distributed estimation, camera network, naive node, consensus

1. INTRODUCTION

In recent years, multi-camera installations have gained immense popularity in various applications. These cameras are usually connected through a network to a centralized data processing unit. But, as the network size increases, the requirement for network connections and processing power on the centralized unit grows rapidly. Also, failure of the central node dictates the failure of the entire system. Decentralized and distributed schemes can be used to cope with these problems. Additionally, distributed vision systems can enhance or enable operations in places without a pre-existing communication infrastructure, like search and rescue operations or remote locations. Distributed algorithms are naturally appropriate for smart sensor networks where each node does the processing of its raw observation to generate processed measurement vectors.

Consensus algorithms are one of the many types of distributed algorithms where each node communicates with its neighboring nodes with its own state information. Then each node updates its own states using information from its neighbors' states and by doing so iteratively, all the nodes can come to a consensus. The consensus is usually a function of the states. Thus, many centralized multi-observation estimation tasks can be implemented in a distributed scenario using consensus algorithms.

In some practical cases, a node in a sensor network may not observe a target (due to limited field of view (FOV) or occlusion). Moreover, due to measurement noise, partial occlusion and directionality of a sensor, the measurements in some nodes can be poor (high measurement covariance). In a consensus based approach, a node shares its measurements only with its immediate network neighbors. Thus, a node has direct access only to the measurements in its *local neighborhood* (consisting of the node and its immediate network neighbors). If in a node's local neighborhood, there are only poor or no measurements about a target, the node becomes *naive* about that target. We will refer to this issue of having insufficient measurement information in a node's local neighborhood as the *naivety* of a node.

In sensor networks, estimating the state from multiple measurements is necessary in many applications. The presence of naive nodes makes the distributed estimation task more challenging which motivates us to propose a generalized consensus algorithm, applicable in any sensor network which is particularly important for camera networks. In this paper, we seek the distributed implementation of the maximum likelihood (ML) estimate of the state of a target viewed by a camera network in this general scenario, accounting for the presence of naivety.

1.1. Related works

Multi-camera networks is a rapidly growing field with applications to security, smart homes, multimedia, and environmental monitoring. A detailed review of the fundamentals, algorithms and applications of multi-camera networks can be found in [1, 2].

Distributed algorithms have been applied to camera networks. In [3], game theory-based distributed optimization algorithms for dynamic camera network reconfiguration and consensus algorithms for scene analysis was proposed. Distributed tracking approaches were described in [4]. In [5], the authors showed how centralized algorithms based on linear al-

This work was partially supported under ONR grant N00014-09-1-0666.

gebraic operations such as SVD, least squares, PCA, GPCA, 3-D point triangulation, pose estimation and affine SfM can be made distributed by using simple distributed averages. In [6, 7] distributed multi-camera tracking schemes were proposed.

One of the popular distributed estimation approaches is based on consensus schemes [8]. In a distributed multi-agent system, different agents may propose different descriptions of an observed target, and agents might need to reach to a *consensus*. There has to be a protocol (called *consensus algorithm*) run by each agent that makes all the agents in the network reach a *consensus*. The *consensus* they try to reach is usually a function of their initial proposals. For example in [9], the average consensus algorithm is proposed which can compute the arithmetic mean of the initial proposals in a distributed manner.

2. PROBLEM FORMULATION

Consider a sensor network with N cameras. The communication in the network can be represented using an undirected graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$. The set $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_N\}$ contains the vertices of the graph and represents the cameras. The set $\mathcal{E}$ contains the edges of the graph which represents the available communication channels between different cameras. Also, let $\mathcal{N}_i$ be the set of cameras having a direct communication channel with camera $\mathcal{C}_i$ (shares an edge with $\mathcal{C}_i$).

In this paper, we are interested in estimating each targets' current state. For example, a target's state can be its position and velocity or pose with respect to the global reference frame. Let us denote the state vector as $\mathbf{x} \in \mathcal{R}^p$. For a position estimation task, this state vector might be the 2d position and velocity on the ground plane. Each node processes its raw observations and gets a noisy measurement that is a function of the state $\mathbf{x}$. For a position estimation task, a measurement might be the position of a target in a camera's pixel coordinate system[1]. The processed measurement $\mathbf{z}_i$ of a target at node $\mathcal{C}_i$ can be expressed using the following observation model,

$$\mathbf{z}_i = \mathbf{H}_i\mathbf{x} + \nu_i \tag{1}$$

Here $\mathbf{z}_i \in \mathcal{R}^m$, $\mathbf{H}_i$ is the observation matrix for node $\mathcal{C}_i$ and measurement noise ν_i is assumed to be $\mathcal{N}(\mathbf{0}, \mathbf{R}_i)$. We will use the information form of the estimators. Thus, we will mostly operate on inverse covariance matrices (also known as information matrices). Note that the information in $\mathbf{z}_i$ is $\mathbf{W}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i$ and the information form of the measurement is $\mathbf{y}_i = \mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{z}_i$. As consensus algorithms require each node to have an initial state, if a camera does not have a measurement, we need to initialize $\mathbf{z}_i$ to some value. In such case, we will use a zero vector for $\mathbf{z}_i$ and $\mathbf{R}_i^{-1}$ will be set to a zero matrix (due to zero information content).

[1] In this paper, we intend to solve only the state estimation task, assuming that the data association for different targets (in case of multiple targets) is provided.

The collection of all measurements of a target from all cameras can be expressed as,

$$\mathbf{z}_c = \mathbf{H}_c\mathbf{x} + \nu_c \tag{2}$$

Here, $\mathbf{z}_c = [\mathbf{z}_1^T, \mathbf{z}_2^T, \dots \mathbf{z}_N^T]^T$ is the stack of all measurements in the network. $\mathbf{H}_c = [\mathbf{H}_1^T, \mathbf{H}_2^T, \dots \mathbf{H}_N^T]^T$ is the stack of all the observation matrices.

We assume the measurement noise to be uncorrelated across cameras. Thus, the measurement covariance matrix will be block diagonal as,

$$\mathbf{R}_c = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & & \\ \vdots & & \ddots & \\ \mathbf{0} & \dots & & \mathbf{R}_N \end{bmatrix} \tag{3}$$

Here, $\mathbf{R}_i \in \mathcal{R}^{m \times m}$ and $\mathbf{R}_c \in \mathcal{R}^{Nm \times Nm}$.

Next, we will review the average consensus algorithm which will be utilized in the distributed implementation of our proposed estimator.

2.1. Average consensus: Review

Average consensus [9] is a popular distributed algorithm to compute the arithmetic mean of some values. Suppose, we have some values $\{x\}_{i=1}^N$ at N nodes and we are interested in computing the average of these values i.e. $\frac{1}{N}\sum_{i=1}^N x_i$, in a distributed manner. Here, x_i can be a scalar, a vector or a matrix.

In average consensus algorithm, each node initializes its consensus state as $x_i(0) \leftarrow x_i$ and runs the following protocol iteratively

$$x_i(k) = x_i(k-1) + \epsilon \sum_{j \in \mathcal{N}_i} (x_j(k-1) - x_i(k-1)). \tag{4}$$

At the beginning of iteration k, a node $\mathcal{C}_i$ sends its previous state $x_i(k-1)$ to its immediate network neighbors $\mathcal{C}_j \in \mathcal{N}_i$ and similarly receives the neighbors' previous states. Then it updates its states using (4). By iteratively doing so, the values of the states at all the nodes converge to the average of the initial values. Here ϵ is the rate parameter which should be chosen between 0 and $\frac{1}{\Delta_{max}}$, where Δ_{max} is the maximum degree of the network graph $\mathcal{G}$. More information about average consensus and about the rate parameter ϵ can be found in [9].

Next, we will derive our distributed maximum likelihood estimation (DMLE) algorithm utilizing this consensus scheme.

3. DISTRIBUTED MAXIMUM LIKELIHOOD ESTIMATION (DMLE)

In this section, first we will state the maximum likelihood estimator for our problem in a centralized scenario and then derive the distributed implementation of it.

The centralized maximum likelihood estimate of $\mathbf{x}$ is given by,

$$\hat{\mathbf{x}}_{ML} = (\mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{H}_c)^{-1} \mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{z}_c. \tag{5}$$

As, $\mathbf{R}_c$ is block diagonal, $\mathbf{R}_c^{-1}$ is also block diagonal. Thus, we can write,

$$\mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{H}_c = \sum_{i=1}^{N} \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i = \sum_{i=1}^{N} \mathbf{W}_i \tag{6}$$

and,
$$\mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{z}_c = \sum_{i=1}^{N} \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{z}_i = \sum_{i=1}^{N} \mathbf{y}_i \tag{7}$$

Now, using (6) and (7) in (5) we get,

$$\begin{aligned} \hat{\mathbf{x}}_{ML} &= \left(\sum_{i=1}^{N} \mathbf{W}_i\right)^{-1} \sum_{i=1}^{N} \mathbf{y}_i \\ &= \left(\frac{\sum_{i=1}^{N} \mathbf{W}_i}{N}\right)^{-1} \frac{\sum_{i=1}^{N} \mathbf{y}_i}{N} \end{aligned} \tag{8}$$

and,
$$\begin{aligned} Cov(\hat{\mathbf{x}}_{ML}) &= (\mathbf{H}_c^T \mathbf{R}_c^{-1} \mathbf{H}_c)^{-1} \\ &= \left(\sum_{i=1}^{N} \mathbf{W}_i\right)^{-1} \end{aligned} \tag{9}$$

Now, according to the definitions of $\mathbf{W}_i$ and $\mathbf{y}_i$, each node $\mathcal{C}_i$, can compute $\mathbf{W}_i$ and $\mathbf{y}_i$ from its own measurement and model parameters. According to (8), the ML estimate is the ratio of the average $\mathbf{y}_i$ and average $\mathbf{W}_i$. It is theoretically guaranteed that using the average consensus algorithm as described in Sec 2.1, each node can asymptotically compute the average $\mathbf{y}_i$ and average $\mathbf{W}_i$. This, along with the expression in (8) guarantees that we can asymptotically compute the centralized ML estimate of (5) using our proposed framework. As ML is an optimal estimator for our problem, and we guarantee to asymptotically compute the ML estimate, our estimator is also an optimal estimator for the problem. The overall implementation of this method is shown in Algorithm 1.

Thus, in this section, we have derived the DMLE algorithm to estimate in a distributed manner the maximum likelihood of the state of a target at each node. We also theoretically guaranteed that our algorithm is an optimal estimator which asymptotically converges to the centralized ML estimate of (5). In the next section, we will experimentally evaluate our method.

4. EXPERIMENTS

To validate our proposed approach for distributed maximum likelihood estimation, we carried out a simulation in a 500x500 unit area. The area was covered with 5 cameras where each can observe a certain portion of the entire area. The FOVs of the cameras are shown in blue triangles in Fig 1. It was assumed that the cameras were connected using a

Algorithm 1 DMLE at node $\mathcal{C}_i$ when new measurement arrives

Input: Observation matrix $\mathbf{H}_i$, measurement $\mathbf{z}_i$, measurement covariance matrix $\mathbf{R}_i$, rate parameter ϵ and number of iterations K.

1) Compute initial fused information matrix and vector

$$\mathbf{W}_i(0) = \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \tag{10}$$

$$\mathbf{y}_i(0) = \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{z}_i \tag{11}$$

2) Perform average consensus on $\mathbf{W}_i(0)$ and $\mathbf{y}_i(0)$ independently

for k = 1 to K **do**

a) Send $\mathbf{W}_i(k-1)$ and $\mathbf{y}_i(k-1)$ to neighbors $j \in \mathcal{N}_i$

b) Receive $\mathbf{W}_j(k-1)$ and $\mathbf{y}_j(k-1)$ from neighbors $j \in \mathcal{N}_i$

c) Update:

$$\mathbf{W}_i(k) = \mathbf{W}_i(k-1) + \epsilon \sum_{j \in \mathcal{N}_i} (\mathbf{W}_j(k-1) - \mathbf{W}_i(k-1)) \tag{12}$$

$$\mathbf{y}_i(k) = \mathbf{y}_i(k-1) + \epsilon \sum_{j \in \mathcal{N}_i} (\mathbf{y}_j(k-1) - \mathbf{y}_i(k-1)) \tag{13}$$

end for

3) Compute ML estimate and Information matrix

$$\hat{\mathbf{x}}_i = \mathbf{W}_i(K)^{-1} \mathbf{y}_i(K) \tag{14}$$

$$Cov(\hat{\mathbf{x}}_i) = (N\mathbf{W}_i(K))^{-1} \tag{15}$$

4) Output: ML estimate $\hat{\mathbf{x}}_i$ and covariance $Cov(\hat{\mathbf{x}}_i)$.

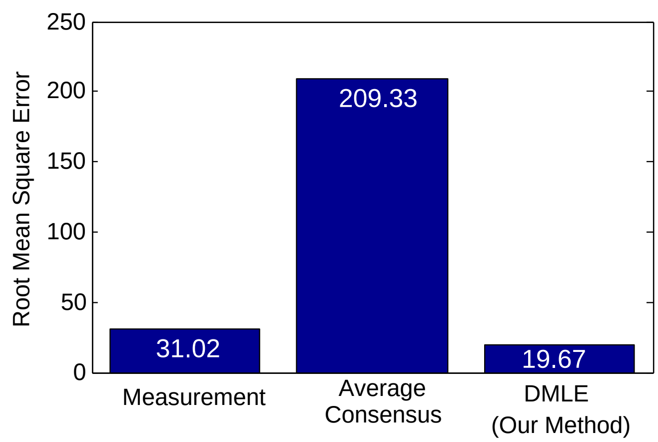

Fig. 2. In this figure, the root mean square error (RMSE) for different methods averaged over 10000 simulation runs are shown. The first bar is the RMSE for the individual measurements. The second and third bars shows the RMSE of the estimates for both the average consensus and DMLE methods. This clearly shows that the RMSE is much less in our method than the average consensus algorithm and is also less than the individual measurements (first bar).

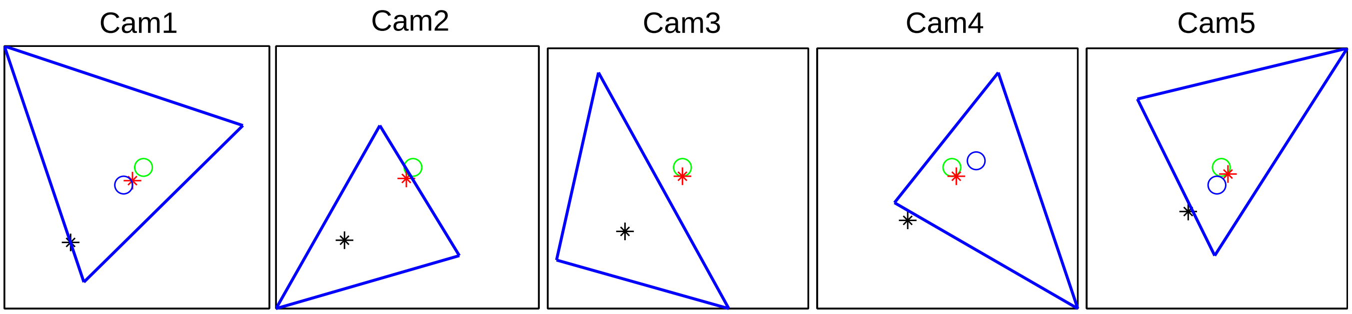

Fig. 1. In this figure, an example from our simulation is shown. The region of surveillance is drawn five times, once for each camera. Here, the green circles (○) denote the ground truth position and the blue circles (○) denote the observations at each camera when the target is in the camera's FOV (blue triangle). The black asterisks ($*$) denote the position estimates from the average consensus algorithm and the red asterisks ($*$) denote the estimates from our proposed algorithm (DMLE). From the figure it is evident that the DMLE estimates ($*$) are much closer to the ground truth (○) than the average consensus estimates ($*$).

peer-to-peer topology as $C_1 \leftrightarrow C_2 \leftrightarrow C_3 \leftrightarrow C_4 \leftrightarrow C_5$. A target appeared at random locations in the area where at least one camera could observe it. A Gaussian noise of $\mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$ was added to the actual position to generate each observation. α was randomly chosen from 1 to 1000 at different simulation runs for different cameras. Total number of iterations K, was set to 10 and the consensus rate parameter ϵ, was set to 0.2.

For a particular simulation run, at first the position was estimated as the average of the measurements (not considering the measurement covariances) using average consensus algorithm. Then it was separately estimated using the proposed DMLE algorithm.

In Fig 1, an example from our simulation is shown. From the figure it is evident that the DMLE results are much closer to the ground truth than the average consensus results. In a consensus scheme, it is necessary that each node has its own initial state. Thus, if the measurement covariances are not incorporated in the estimation process, the nodes which had no measurements, introduced a bias towards the arbitrarily set measurement values in such nodes. In contrast to this, our maximum likelihood estimator in (5) is an unbiased estimator, which is essentially computing the weighted average instead of the average. In Fig 2, the root mean square error (RMSE) for different methods are shown. This clearly shows our method outperforms other methods.

5. CONCLUSION

In this paper, we have presented a novel algorithm (DMLE) to compute the maximum likelihood estimate of the state of a target in a camera network in a distributed framework. Theoretical guarantee of optimality and convergence was provided for the method. Experimental results were provided to compare the estimation accuracy with other methods which showed that our method performed much better than other methods.

6. REFERENCES

[1] H. Aghajan and A. Cavallaro, *Multi-Camera Networks: Principles and Applications*, Academic Press, 2009.

[2] B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds., *Distributed Video Sensor Networks*, Springer London, 2011.

[3] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed Wide Area Scene Analysis in Reconfigurable Camera Networks," in *Signal Processing Magazine*, 2011.

[4] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *Signal Processing Magazine, IEEE*, 2011.

[5] R. Tron and R. Vidal., "Distributed algorithms for camera sensor networks," *IEEE Signal Processing Magazine*, 2011.

[6] A. T. Kamal, C. Ding, B. Song, J. A. Farrell, and A. K. Roy-Chowdhury, "A generalized kalman consensus filter for wide area video networks," in *Controls and Decision Conference*, 2011.

[7] B. Song, A. T. Kamal, C. Soto, C. Ding, A. K. Roy-Chowdhury, and J. A. Farrell, "Tracking and Activity Recognition Through Consensus in Distributed Camera Networks," in *IEEE Trans. on Image Processing*, 2010.

[8] R. Olfati-saber, J. A. Fax, and R. J. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE*, 2007.

[9] R. Olfati-Saber and R. J. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference*, 2003.