Sensor Synchronization for Energy Efficient Multiple Object Tracking

Fan Zhou*, Goce Trajcevski†, Besim Avci†, Peter Scheuermann†
*School of Computer Science and Engineering
University of Electronic Science and Technology of China
Email: fan.zhou.uestc@gmail.com
†Electrical Engineering and Computer Science Department
Northwestern University
Email: goce,besimal,peters@eecs.northwestern.edu

***Abstract*—This work addresses the problem of synchronizing the sensors involved in the task of multiple object tracking (MOT) in Wireless Sensor Networks (WSN). We aim at reducing the overall in-network energy consumption along with bounding the uncertainty regarding targets' locations in WSN. Designing energy efficient scheduling mechanism is a challenge in WSN tracking scenarios due to the limitations on target's movement prediction, and lack of global network knowledge. The main observation of this work is that task conflicts and channel congestion preclude the utilization of the nodes shared by *common tracking tasks*, which may result in poor Quality of Tracking (QoT) and/or increasing target ambiguity. In order to tackle this problem, we propose a lightweight sensor scheduling policy – *Synchronization based Sampling Reduction (SSR)*, which explicitly prunes the redundant measurements in the conflicting nodes without decreasing QoT, through synchronizing the tracking tasks. In addition to conserving the energy by reducing the samplings, SSR also is capable of mitigating the data associating problem in MOT, without requiring any global knowledge about the network. Our experiments demonstrate that SSR can significantly reduce the number of locations sampling, when compared to naïve approach that does not coordinate the nodes involved in multiple object tracking.**

I. Introduction

Technological and manufacturing advances in the recent years have enabled the development of relatively cheap sensor nodes that can be deployed in various environments and can self-organize in a network for the purpose of achieving a desired task. Application domains have spanned from environmental (e.g., animal tracking, phenomena monitoring), through safety and disaster remediation (e.g., fire detection and tracking) to health and traffic management.

Among the canonical problems in Wireless Sensor Networks (WSN) context applicable to scenarios ranging from intruder detections to animal surveillance is the one of *tracking* moving objects. A number of research works have addressed diverse facets of the problem of moving object tracking in WSN like, for example, optimal sensor deployment patterns [2], [13] and methodologies on improving localization accuracy [1], [22]. Given that in typical scenarios the sensor nodes are powered by batteries which are non-rechargeable, prolonging the operational lifetime of the nodes is a paramount. Towards that, many works have proposed tracking schemes that have specifically focused on the energy-efficiency aspect [6], [8], [9], [11], [14], [20].

This work was supported by the U.S. National Science Foundation under Grant CNS-00910952.

One important aspect of the power-conservation is the orchestration of the *sleeping* schedule of the nodes that participate in detecting the location of a given moving object. To devise an efficient sleep/wakeup strategy, the nodes are organized into discrete clusters based on the geographical information and the movement of target. Thereby only the sensors required to participate in location-detection are activated (cf. [19]). Clearly, letting some of the nodes sleep will impose certain trade-offs with respect to the tracking quality – and the problem of balancing it with the energy consumptions via dynamically clustering the nodes along the target's trajectory has been addressed in [9], [10], [8]. In our recent work [21] we devised a strategy to divide the tracking task in discrete *tracking epochs*, during which a subset of sensors are selected to cover the target's *possible-locations distribution area* called *uncertain disk*, as another attempt to conserve the in-network energy consumption.

However, above mentioned schemes may fail when extending to *multiple object tracking* (MOT) settings. One of the difficulties that arise is known as the *data association problem*. Namely, it is hard to discriminate the different moving objects based on the location-measurements when there exist two or more of such objects occurring in the sensing area of a single tracking node. Many works addressed the data association ambiguity, and two significant examples are: MHT (*multiple hypothesis tracking* [18]) and JPDAF (*joint probability data association filter* [5]). Recently, Yeow et al. [20] formulated the MOT problem as a hierarchial Markov decision process (MDP), and solved it through Q-learning algorithm. Similarly, the authors in [6] derives suboptimal solutions for MOT which is formulated as partially observable MDP.

We target a very specific aspect of the problem energy-efficient tracking in MOT settings: how to synchronize the sensors' scheduling, especially the ones participating in tracking two or more objects. We also investigate what the benefits

978-1-4673-0390-3/12/$31.00

and what the trade-offs of such sensor synchronization are. The motivation for this work is illustrated in Figure 1, which shows the motion plan (trajectory) of two objects O_1 and O_2, being tracked by sensors in a given area. To reduce the sensing and communication overhead, sensors located in the overlapping area (red sensors in Figure 1) should have their individual tracking tasks orchestrated in a collaborative manner to minimize the number of sleep-to-wakeup transitions. However, the beginning time and task duration required by tracking different objects need not be consistent – e.g., task duration is typically inverse to the velocity of targets. This situation will be even worse when taking the asynchronous process into account. Suppose that tracking O_1 requires the nodes to wake up at t_k while tracking O_2 requires them to wake up at $t_k + \epsilon (> 0)$, and let τ denote the execution time for tracking O_1 (including sensing and communication), the tracking task for O_2 would fail to read the channel if $\epsilon < \tau$. Thus, exploiting the shared nodes may result in channel congestion, which requires process synchronization and sensor task rescheduling. Figure 1 shows the tracking request and scheduling table for nodes in the two clusters and, to re-iterate – we are interested in planning the tasks of the red nodes.

The task of coordinating the exploitation of such nodes is nontrivial, and a peculiar difficulty is due to the many possible outcomes for the data association. Suppose there exists q objects that can be simultaneously sensed by node S_i at each sampling step. Without considering the false alarms, S_i generates q measurements and thus $q!$ possible data associations. The computational complexity increases exponentially to $(q!)^p$ if there are p sampling times over a time window – hence, decreasing the value of p is of a practical interest.

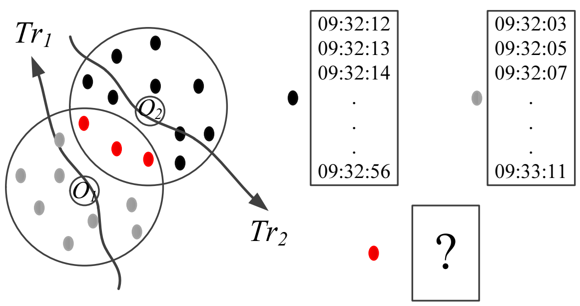


Fig. 1. Illustration of Motivation.

Our goal is to exploit the knowledge that some nodes may be "shared" in MOT tasks for the purpose of reducing their energy consumption. Specifically, we focus on designing a policy to rearrange their schedules in the common tracking tasks. The main contribution of this work is a scheduling scheme – *Synchronization based Sampling Reduction (SSR)*, which improves the energy efficiency in MOT settings via pruning the redundant measurements. SSR works in a simple but efficient way by synchronizing the tracking tasks on a given sensor node, and is a lightweight scheme that can operate in a distributed manner.

The rest of this paper is organized as follows. In Section II, we describe the basic network settings and give some necessary background. Section III presents the details of the SSR scheme for multiple object tracking. Experimental observations illustrating the advantages of SSR is presented in section IV. We review the related works on multiple object tracking in section V and we give the concluding remarks, along with possible future extensions in Section VI.

II. Preliminaries

We consider a sensor network $\mathbf{S}$ consisting of N sensor nodes, randomly and densely deployed over a 2D field $\mathbb{R}$. Each node $S_i \in \mathbf{S}$ can detect whether a moving object locates in its sensing range R_s, via some range based methods. Furthermore, S_i is aware of its location as well as the locations of its one-hop neighbors $NB(S_i)$, via GPS or some localization techniques [3]. We assume that the transmission range of each node, R_c, is twice as large as its sensing range, R_s (i.e., $R_c = 2R_s$) – thus, sensing coverage suffices for connectivity. Furthermore, each node maintains a local clock and is capable of rescheduling its sampling tasks.

The trajectories of q moving objects $O_1, ..., O_q$ to be tracked are assume to be independent from each other. The tracking nodes are organized in clusters throughout *tracking epochs*, during which a given node in each cluster is elected as its *tracking principal* S_p^k, acting as temporal data fusion center for object O_k. We say that two moving objects O_m and O_n have "contact" with each other at time t, whenever there exists sensor $S_i \in \mathbf{S}$, such that (the locations of) both O_m and O_n can be detected by S_i at t. In other words, O_m contacts O_n *iff* $||S_i, L_m^t|| \leq R_s$ and $||S_i, L_n^t|| \leq R_s$ where L_*^t denotes the position of target O_* at t. Nodes given cluster, formed throughout a particular sequence of epochs, have no global knowledge of schedules of nodes in other clusters.

For a particular object O_i, if it does not contact with other objects, the sensor scheduling is performed in the manner of *location uncertainty bounded tracking (UBT)* [21]. The basic idea of UBT is that the tracking task is organized in discrete *tracking epochs*, during which a subset of all detecting sensors are sufficient for the purpose of monitoring the moving object while bounding the uncertainty regarding the position of target in a circle area, called *uncertain disk* $D(r_u)$. Specifically, assume the current location of target is L_t and the closest node is S_p which is selected as the tracking principal, the *tracking epoch* $\mathcal{E}$ is given by:

$$\mathcal{E} = \lfloor \frac{r_u}{V_{max}} \rfloor \tag{1}$$

where r_u denotes the radius of uncertain disk and V_{max} is the maximal velocity V_{max} which is the only knowledge regarding moving objects. Given the desiderata of maintaining communication between the S_p and $NB(S_p)$ we have the constraint: $r_u = R_c - ||S_p, L_t||$.

Parameter r_u specifies the area to be covered during $\mathcal{E}$ throughout which the sampling task and target localization are saved, if the uncertainty w.r.t. target position within $D(r_u)$ can be tolerated. We derived two heuristics in [21] – GUMO

(*Greedy Uncertain Moving Object coverage sensor set selection*) and PAB (*PAttern Based coverage sensor set selection*), to select sensors during each epoch $\mathcal{E}$ for tracking of a single object.

Tracking single object when UBT is applied provides significant energy savings as demonstrated in [21]. In this work we target at MOT through designing energy efficient sampling schemes for tracking sensors, especially for those shared by multiple tasks.

III. Multiple Object Tracking with Sensor Synchronization

We now proceed with presenting the details of the *Synchronization based Sampling Reduction (SSR)* scheme designed for multiple object tracking. Without loss of generality, we denote the targets as $O_1, ..., O_q$ and corresponding trajectories $Tr_1, .., Tr_q$ according to the order of the tracking request being tasked to a given sensor S_i. In addition, we denote the sampling frequency and the duration of the respective interval of S_i as f_i and I_i. Moreover, we denote the beginning and ending times of a given scheduling period pertaining to the tracking target O_i with t_s^i and t_e^i, respectively.

A. Synchronization based Sampling Reduction

The main objective of SSR is to reduce the samplings and resolve the scheduling conflicts in those sensors shared by multiple tracking clusters through synchronizing their sensing tasks. Since different objects moves in different patterns i.e., velocity and direction, in order to guarantee a certain tracking quality, the sampling frequency is typically in inverse ratio to object's displacement per interval [8].

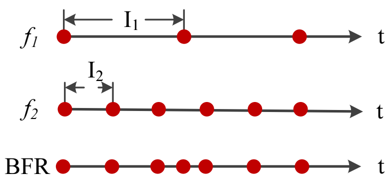


Fig. 2. Combination of Samplings.

Suppose that S_i is required to track two objects with sampling frequency f_1 and $f_2(> f_1)$ during a period of time. The brute force scheme (heretofore abbreviated as BFR) would wake up the sensor node S_i at sampling steps obtained as a combination of f_1 and f_2, which exhibits an irregular sampling intervals – as illustrated in Figure 2. Obviously, BFR may yield better tracking quality due to the higher sampling frequency, however, this incurs a higher energy consumption. Just as importantly, redundant measurements bring a higher overhead of the object identity discrimination due to the data association problem.

SSR operates in a simple but efficient way to reduce samplings while guaranteeing the coverage of target. Specifically, it consists of the following steps:

1) Initially, S_i operates in a low-power sleeping mode if there is no incoming tracking task; however, once it is selected to cover O_1 starting at t_s^1, it converts to tracking mode. During tracking epoch $\mathcal{E}_1$, S_i samples in frequency f_1 and transmits samplings to tracking principal S_p^1.
2) If no other tracking task is needed during $\mathcal{E}_1$, S_i restores to sleeping mode at $t_s^1 + \mathcal{E}_1$; however, whenever a request of tracking another target O_2 takes place, SSR scheme is launched, during which two cases exist:
   - **Case 1.** $f_1 \leq f_2$ ($I_1 \geq I_2$). If $t_e^2 \geq t_e^1$, S_i maintains its sampling cycles (f_1) until physical time t_e^1 reached, after which its sampling frequency transforms to f_2, cf. figure 3(a); otherwise, S_i increases a sampling step at t_e^2 based on frequency f_1, as illustrated in figure 3(b).
   - **Case 2.** $f_1 > f_2$ ($I_1 < I_2$). S_i immediately sets its sampling frequency as f_2 until t_e^* is reached, which is determined by $\min\{t_e^1, t_e^2\}$. If $t_e^* = t_e^2$ (Case 2", cf. Figure 4(b)), S_i changes its sampling interval back to f_1 after t_e^*; otherwise, it increases a sampling at t_e^1 based on f_2, as shown in Figure 4(a).
3) If there exits tracking tasks for $O_3, ..., O_n$ before t_e^*, S_i sets its frequency to $f_* = \min\{f_3, ..., f_n\}$ and operates in the same way as stated above.

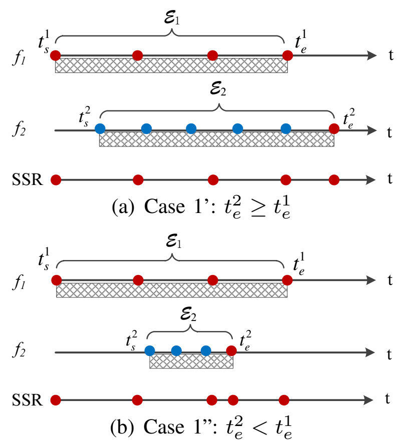
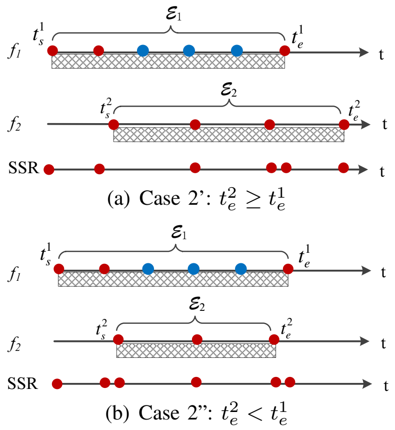


Fig. 3. Case 1, $f_1 \leq f_2$.

In general, SSR is prone to sample in a lower frequency in order to conserve energy and mitigate the data association problem. The main benefit of SSR is sampling reduction, e.g., the blue sampling points in Figure 3 and 4 are discarded.

Another advantage of SSR is that it has the potential of mitigating data association problem, which is generated by the shared sensor nodes without false alarms and missed detections. To elaborate, consider Case 1' and suppose during (t_s^2, t_e^1) both targets O_1 and O_1 can be sensed by sensor S_i. At each sampling step, S_i observes two measurements m_A and m_B; however, two possible associations exist since S_i cannot distinguish which one is generated by O_1 and which one is generated by O_2. Then there are 2^8 possible associations due to 8 samplings in BFR while, however, there exist only 2^3 possible associations with SSR during (t_s^2, t_e^1).



(a) Case 2': $t_e^2 \geq t_e^1$

(b) Case 2'': $t_e^2 < t_e^1$

Fig. 4. Case 2, $f_1 > f_2$.

It is also worthy to note that SSR would not increase the uncertainty regarding the targets compared to single object tracking with UBT scheme. Consider Case 1, where subcase 1' guarantees monitoring O_2 during $[t_s^2, t_e^1]$ before converting to mode f_2. In subcase 1'', if O_2 locates in the sensing range of S_i at t_e^2, SSR is capable of measuring it; otherwise, O_2 is promised to be sensed by other node(s) in tracking cluster selected by GUMO, if the uncertain disk is totally covered. The similar situation also holds in Case 2.

B. Object Identification

The main objective of this paper is to coordinate the sampling reduction and sensor synchronization. While we are not formally addressing the problem of the identity management in MOT settings, we conclude this section with outlining a possible method for incorporating that aspect into the SSR scheme.

The sampling data of each sensor is a tuple indicating the measurement value, sampling time and an indicator describing each object, e.g., the range measurements (m_1, t_i, O_1) and (m_2, t_i, O_2) correspond to the times when two objects are well-separated, as illustrated by the bottom portion of Figure 5. This type of data is transmitted to the tracking principals of each cluster, which can either directly forward it to a given sink, or perform some aggregation and analysis of the moving patterns. Subsequently, the tracking principal hands off the data to the subsequent principal along the target trajectories.

Once the uncertain disks of two objects overlap with each other, which is when the SSR policy is triggered, the sampling data associated with shared nodes introduces the identity ambiguity. In accordance with the works on disjunctive logic programming [16], we propose to introduce the uncertain notation as follows: we let $(m_1 \vee m_2,\, t_j,\, O_1 \vee O_j)$ denote a tuple which indicates that the measurements at t_j resulted in two location-values m_1 and m_2, and there are two objects – O_1 and O_j – that could qualify for the location-to-time mapping, but association is uncertain. Without enough mobility information required to clarify the object ambiguity, the tracking principals will carry the disjunctive tuples.

We note that in certain practical settings the target identity could be post-determined with the help of the velocity data or other estimation techniques [20], [6]. For example, during time period (t_j, t_k), the two targets respectively exhibit average velocity $V_1 = 3m/s$ and $V_2 = 7m/s$ – if both O_1 and O_2 do not contact other objects during (t_j, t_k), then we can prune some possible indicators, and distinguish the objects. However, we re-iterate that investigating this topic in detail is a subject of our future work.

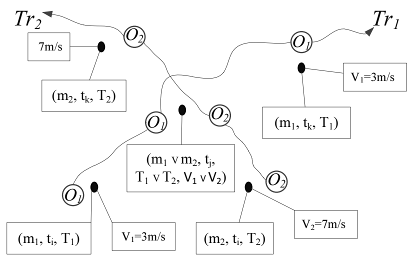


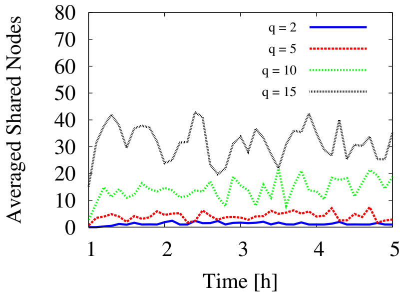
Fig. 5. Illustration of Identifying Objects.

IV. Experimental Observations

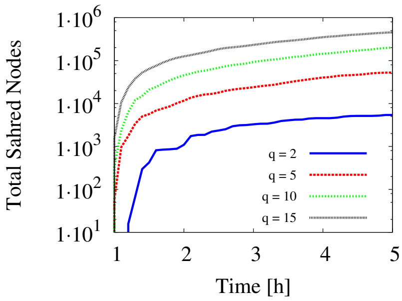
To evaluate the performance of SSR scheme, we compare it with a basic tracking scheme (BFR), as stated in section III, i.e., awakening the nodes at scheduled sampling steps for all tracking clusters. The experiments were conducted on the open source simulator for WSN, SIDnet-SWANS [7]. Each run simulates 700 homogeneous sensor nodes configured with following specifications: (1) 38.4kbps radio data rate on the MAC802.15.4 protocol; (2) energy consumption characteristics of Mica2 Motes; (3) 5 hours spanning time including 1 hour of network bootstrap and configuration, and 4 hours of target tracking time.

We simulate q (= 2, 5, 10, 15) moving objects according to the RWP model with velocity randomly distributed in range [4mph, 25mph] – lower and upper bound respectively simulates the average velocity of walking people and driving cars. Nodes' sampling frequency f in each cluster is proportional with the velocity of the target. We only focus on the behavior of the nodes shared by multiple tracking tasks and ignore other tracking nodes.The results presented are obtained by averaging over 10 runs. Note that we advisedly invalidate GUMO selection here in order to study the effects of deploying SSR scheme independently.

To reschedule the nodes in the overlapping uncertain disks which are generated when two or more targets "meet" with each other(s), first we need to understand the characteristics of object contacts, e.g., time, probabilistic and frequency. It has been demonstrated in literature [4] that the inter-meeting time between two independent mobile objects decays at least exponentially fast for RWP or Random Walk Mobility model given the boundary is finite. We simulate a MOT scenario with q objects moving in RWP pattern. Figure 6(a) and 6(b) respectively shows the average and accumulated number of nodes participating in multiple tracking tasks in 5 hours simulation time, which reveals the contact frequency of multiple objects in MOT. It can be seen that the shared nodes increase largely with the number of objects. Thus, in MOT setting with a higher value of q, significant energy is expected to be saved by trading off some samplings in the shared nodes, which motivates this work.



(a) Average Number of Shared Nodes



(b) Total Number of Shared Nodes

Fig. 6. Contact frequency. $N = 700$.

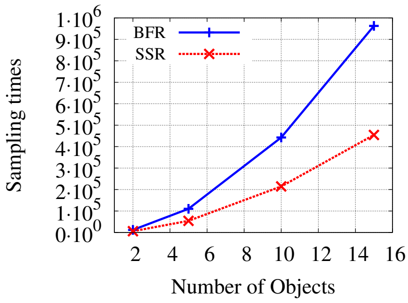
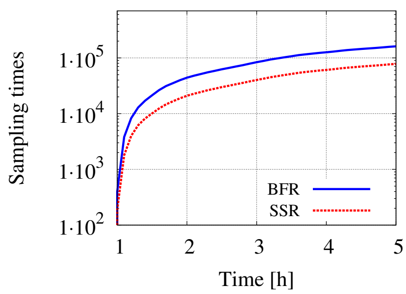


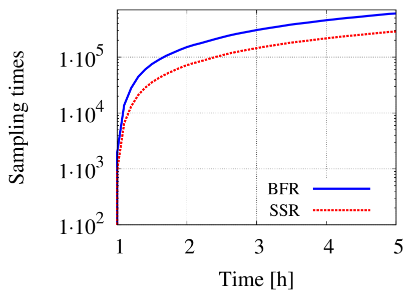
Fig. 7. Impact of q.

We first investigate the impact of the parameter q on the number of samplings. Obviously, the more of the targets, the higher probability of contacting between them, and hence more shared nodes required to participate in tracking (c.f Figure 6(b)). As in Figure 7, there are few sampling times of both schemes, and their difference is not apparent when target number is small ($q = 2$); however, sampling times of two schemes increase significantly with the value of parameter q. At the same time, SSR is capable of controlling the size of samplings in a lower level compared to BFR.

Figure 8 illustrates the impact of the nodes density λ on the performance of two methods regarding sensor sampling times, where SSR outperforms BFR in both settings. Robust tracking requires denser nodes deployment which, however, may increase the overhead of communication and sensing. In MOT scenario, higher node density may also increase the impact of the data association problem and, consequently, the resources' consumption.



(a) $\lambda = 10$



(b) $\lambda = 20$

Fig. 8. Samples over time.

Finally, Figure 9 shows the effect of SSR scheme on sampling reduction. For lower value of q, say 2 objects, the advantage of SSR is trivial which is a natural result due to the number of sensor synchronization tasks is small. However, a huge amount of sampling reduction would be achieved for higher value of q, say, 15 objects. This result proves our motivation that significant energy savings is expectable when tracking a larger number of objects through sensor synchronization.

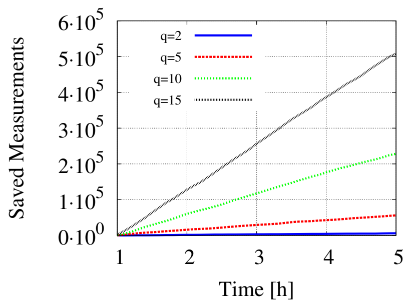


Fig. 9. Size of sampling reduction.

V. Related Work

Energy consumption for MOT is orders magnitude higher than spent on single object tracking, due to more intensive task of sensing, computation and communication. To conserve network resource, tracking multiple objects is normally decomposed into two phases – single object tracking stage if objects are separated, and multiple tracking stage when target contacts happen, e.g., [15]. Recently, energy efficient MOT is studied in [20], [6] where Markov decision process based algorithms are utilized for predicting target trajectory and deriving sensor sleeping strategies. However, such protocols rely heavily on the prediction accuracy and operate in a central manner. While SSR is a sensor scheduling approach targeting for MOT problem, it focuses on a synchronization of the sensing tasks via coordinating the sleeping schedule of a *specific subset* of the nodes involved in tracking. It is a lightweight solution that can be easily incorporated into existing moving object tracking protocols.

A number of methodologies have been proposed addressing various aspects of the problem of multiple object tracking, among which data association is a major challenge. MHT [18] and JPDAF [5] are two influential algorithms tackling the data association problem but suffer from high computational complexity of their optimal solutions. Particle filtering provides an alternative way for recursively estimating target state and searching the space of hypotheses (cf. [12]). In a similar sprit, MCMC driven methods such as MCMCDA [17] solve the data association problem through constructing a Markov chain sampling from complex probability distribution. Compared to these works, SSR does not explicitly solve the data association problem; however, it provides a solution on reducing the burden of tackling this problem, i.e., through sampling data reduction on those nodes generating fuzzy measurements.

VI. Concluding Remarks

In this paper, we presented SSR – a task synchronization scheme for multiple object tracking. The proposed scheme exploits the characteristics of the sampling frequency of sensors to prune redundant measurements. Through sampling reduction, not only energy consumption and channel conflicts but also the problem of target ambiguity can be alleviated.

In our future work, we are interested in incorporating more sophisticated models such as Markov decision process to address target state estimation and data association problem. Further, designing energy efficient strategies for multiple target tracking in a more realistic WSN scenario presents a challenge, where constraints such as communication delay, sensing errors and congestion control should be explicitly considered. Another extension of our work is to incorporate the uncertainty of the targets' identities into processing different spatio-temporal queries in WSN settings.

References

[1] A. Baggio and K. Langendoen, "Monte carlo localization for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 718–733, 2008.

[2] X. Bai, C. Zhang, D. Xuan, and W. Jia, "Full-coverage and k-connectivity (k=14, 6) three dimensional networks," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2009, pp. 388–396.

[3] N. Bulusu, J. D. Heidemann, D. Estrin, and T. Tran, "Self-configuring localization systems: Design and experimental evaluation," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, 2004.

[4] H. Cai and D. Y. Eun, "Crossing over the bounded domain: from exponential to power-law inter-meeting time in manet," in *International conference on Mobile computing and networking (MobiCom).* ACM, 2007, pp. 159–170.

[5] I. Cox and S. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 2, pp. 138–150, 1996.

[6] J. A. Fuemmeler and V. V. Veeravalli, "Energy efficient multi-object tracking in sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3742–3750, 2010.

[7] O. Ghica, G. Trajcevski, P. Scheuermann, Z. Bischof, and N. Valtchanov, "Sidnet-swans: a simulator and integrated development platform for sensor networks applications," in *SenSys*, 2008, pp. 385–386.

[8] O. Ghica, G. Trajcevski, F. Zhou, R. Tamassia, and P. Scheuermann, "Selecting tracking principals with epoch awareness," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '10. New York, NY, USA: ACM, 2010, pp. 222–231.

[9] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 129–143.

[10] G. He and J. C. Hou, "Tracking targets with quality inwireless sensor networks," in *Proceedings of the 13TH IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 63–74.

[11] J. Hong, J. Cao, Y. Zeng, S. Lu, D. Chen, and Z. Li, "A location-free prediction-based sleep scheduling protocol for object tracking in sensor networks," in *17th IEEE International Conference on Network Protocols*. IEEE, 2009, pp. 63–72.

[12] C. Hue, J. pierre Le Cadre, and P. Prez, "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Trans. on Signal Processing*, 2002.

[13] R. Iyengar, K. Kar, and S. Banerjee, "Low-coordination topologies for redundancy in sensor networks," in *MobiHoc*. ACM, 2005, pp. 332–342. [Online]. Available: http://doi.acm.org/10.1145/1062689.1062732

[14] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 169–180.

[15] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed state representation for tracking problems in sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, ser. IPSN '04. New York, NY, USA: ACM, 2004, pp. 234–242.

[16] J. Lobo, J. Minker, and A. Rajasekar, *Foundations of Disjunctive Logic Programming*. Cambridge, Massachusetts: MIT press, 1992.

[17] S. Oh, S. Russell, and S. Sastry, "Markov chain monte carlo data association for general multiple-target tracking problems," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1. IEEE, 2004, pp. 735–742.

[18] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, 1979.

[19] Y. Xu, J. Winter, and W.-C. Lee, "Prediction-based strategies for energy saving in object tracking sensor networks," pp. 346–357, 2004.

[20] W. Yeow, C. Tham, and W. Wong, "Energy efficient multiple target tracking in wireless sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 2, pp. 918–928, 2007.

[21] F. Zhou, G. Trajcevski, and B. Avci, "Tracking coverage throughout epochs with bounded uncertainty," in *IEEE International Symposium on Networking Computing and Applications*, 2011.

[22] Y. Zhu, S. J. Gortler, and D. Thurston, "Sensor network localization using sensor perturbation," *ACM Trans. Sen. Netw.*, vol. 7, pp. 36:1–36:23, February 2011.