Computing and Informatics, Vol. 29, 2010, 647–662

NONLINEAR TRAJECTORY DISCOVERY OF A MOVING TARGET BY WIRELESS SENSOR NETWORKS

Ahmed M. Khedr, Walid Osamy

Mathematics Department, Faculty of Science
Zagazig University, Zagazig Egypt
e-mail: {amkhedr, w.osamy}@zu.edu.eg

Manuscript received 20 August 2009; revised 20 January 2010
Communicated by Michael M. Richter

Abstract. Target tracking is an important cooperative sensing application of wireless sensor networks. In these networks energy, computing power and communication bandwidth are scarce. In this paper, we consider a randomly deployed sensor network with sensors acting as a set of distributed datasets. Each dataset is assumed to have its local temporal dataset, along with spatial data and the geographical coordinates of a given object. An approach towards mines global temporal patterns from these datasets and to discovers nonlinear trajectories of a moving object is proposed. It is tested in a simulation environment and compared with straightforward method. The results of the experiments clearly show the benefits of the new approach in terms of energy consumption.

Keywords: Global temporal pattern, mining, nonlinear trajectories, target tracking, wireless sensor networks

1 INTRODUCTION

A wireless sensor network (WSN) usually consists of a set of sensors, deployed in hundreds or thousands over a region. Each sensor is capable to measuring acoustic, magnetic, spatial, or seismic data and the wireless interconnection network enables performing distributed computations over the gathered data by individual sensors, make meaningful inferences at the base station, and then send the data to end user for appropriate action. WSNs promise novel applications in several domains such as forest fire detection, battlefield surveillance, or monitoring of human physiological data are only in the vanguard of improvements provided by the WSNs. Sensor nodes can be spread out in a dangerous or remote environment by low flying airplanes or unmanned aerial vehicles that opens up new application fields. One of the main challenges raised by these networks is the fact that they are usually power constrained, since sensing nodes typically exhibit limited capabilities in terms of processing, communication, and especially run on battery power [18]. Sensor networks' power limitation is impacted by the fact that, once deployed, they are often left unattended for their lifetime. Thus, in order to maximize the WSN's operational lifetime energy conservation is of prime consideration in algorithms used for WSN algorithms.

In this paper, we refer to WSN as a set of geographical distributed stand alone sensors that randomly spread over a region to be monitored and are capable of sensing predefined parameters such as temperature, motion and geographical coordinates of objects in its vicinity and is of interest to determine global patterns in a geographical area.

We consider the problem of tracking a target moving in a region populated by sensor nodes that have limited wireless communication capabilities. The sensors have fixed sensing range and the quality of range estimates degrades with distance from the sensor. The best accuracy is obtained if the measurements from all the sensors can be collected and processed together to estimate the target location. Nevertheless, this global collection is not possible due to communication constraints of individual sensors. To accomplish this task efficiently a distributed mechanism is necessary for target tracking.

We present a methodology for mining geographically distributed datasets for spatio-temporal patterns. In order to achieve energy efficiency in WSN, we present a clustering architecture for the sensor nodes, where instead of sending individual raw data from sensors to end users, multiple data items are aggregated as they are forwarded by the WSN. Our main goal is to perform data mining on the data stored in sensor nodes to extract useful information and then to do a further round of exchange of messages between the sensor nodes and the cluster heads (CHs) to discover global spatio-temporal patterns.

Our distributed mechanism can be described as follows: We assume that there is no central authority governing WSN operation. Since information degrades with distance, only sensors sufficiently close to the target share their data. Second, since there is no designated central authority, an arbitrary sensor takes the role of a fusion center (or leader node) for a given time period and the data is processed at that sensor.

The rest of the paper is organized as follows: related research is reviewed in the following section. In Section 3, we give the basic concepts of the problem formulation. A step by step outline of our algorithm is described in Section 4. The simulation of our algorithm is presented in Section 5. Section 6 analyzes the communication requirements of our algorithm. We conclude our work in Section 7.

2 RELATED RESEARCH

During the past few years surveillance and monitoring applications using WSNs has attracted a lot of attention in the research community. This kind of functionalities form a canonical class of applications which can be constructed with WSNs. The work presented in this paper has been inspired by various existing research efforts, and we introduce data mining algorithm to discover temporal movement patterns of object in a WSN.

A number of approaches have been investigated for object localization and tracking. Some existing approaches such as [19] use the sensor nodes only to collect data and do the computation of the event's location at a base station equipped with more computing resources and power (such as a laptop computer). This circumvents the limited computing resources at the individual sensor nodes and the complexity of a distributed algorithm, but creates the strain on the WSN bandwidth and sensor node batteries by the increased traffic between the nodes around the event and the base station. Other approaches observe events through clusters of sensor nodes that are formed during the self-configuration process [26, 27]. Cluster members are able to directly communicate with the CH and send their information to it. The CH collects the data from all the cluster members and computes the event location.

In [5], the authors present Sextant, a unified framework for node and event localization. Sextant is a comprehensive system that derives its effectiveness from integrating negative as well as positive information, representing areas precisely using Bezier curves, transitively disseminating constraints in the presence of uncertainty, and solving the resulting system of constraints using a distributed algorithm. The resulting system is capable of providing probability distributions for event locations, and non-convex area estimates for node locations to higher level applications.

In [3], a technique for clustering homogeneously distributed data in a peer-to-peer WSN environment is described. The proposed technique is based on the principles of the k-means algorithm. It works in a localized asynchronous manner by communicating with the neighboring nodes. The results showed that, in contrast to the case when all the data is transmitted to a central location for application of the conventional clustering algorithm, the communication cost of their approach is significantly smaller. In [23], the authors propose a fully distributed localization scheme that consists of two steps. The first step of distributed election-winner notification algorithm (DENA) determines the closest sensor node to an event and informs all other sensor nodes about the sensor node closest to the event. The intensity based localization algorithm (ILA) provides a signal independent position estimation of the event and is calculated at the closest winner node. A combination of both these algorithms builds a framework to efficiently and accurately detect and localize events. The novelty of their ILA algorithm is that it is independent of signals emitted by an event.

In [24], a location estimation algorithm on a single sensor node equipped with inexpensive directional antennas is proposed and demonstrated by measuring the received signal strength of the transmission peers. The authors provided an algorithm for location estimation of moving targets, individually by the sensor node; and the estimation is computed by selecting two patch antennas using strongest received signal strength (RSS). The final result is obtained by averaging these two estimated locations. We consider the system proposed by Chin-Lung in [24], as we consider sensor nodes to be capable of capturing and storing coordinate information about a moving object. Other aspects in which localization approaches differ are the kind and the number of sensing modalities used, such as light, magnetism, seismic waves, angle of arrival or time of arrival of acoustic waves.

In [9], the authors suggested a target tracking algorithm MCTA using minimal tracking area called tracking contour that is based on the vehicular kinematics. MCTA minimizes the number of working sensor nodes in terms of the communication and sensing energy cost during the mobile target's trajectory. They showed that the ratio of tracking contour's working sensor number to tracking circle's working sensor number is proportional to the ratio of the tracking contour's area to track circle's area. This indicates that the reduction of the tracking area leads to saving in communication and sensing energy. Also, in order to reduce the dissemination of tracking contour information within the tracking contour, they used the RF transmission power control and directional antenna, leading to the minimization of the number of RF receiving sensors.

In [14], the authors proposed a simple scheme for tracking object's movement. The proposed scheme is based on the difference of signal arrival times, which is so-called inter-node time difference-of-arrival (ITDOA). It has a similar measurement characteristic to classical time difference-of-arrival (TDOA) measurement. However, the ITDOA scheme has an advantage over TDOA; it does not require any time synchronization between reference nodes. In context of time synchronization or simply timesync, it is useful for establishing temporal ordering of events (x happened before y) and real-time issues (x and y happened within a certain interval) [20]. Timesync also may be used to coordinate future actions at two or more nodes (x, y, and z will all happen at time T). In mobile target tracking, it is essential that the nodes act in a coordinated and synchronized fashion and global clock synchronization is required, which is all the nodes of the WSN need to refer to a common notion of time. For example, consider the problem of tracking a moving target using proximity sensors, where some sensor nodes are deployed in the environment and their proximity sensors detect when the moving object passes in their vicinity. Assuming that the position of the sensors is known, it is essential that the instants of detection are precisely time-stamped for determining the trajectory (direction and speed) of the moving object. Clearly, the precision of the tracking algorithm based on this system is limited by the accuracy of the clock synchronization [1]. In our algorithm, we implicitly assume that the node clocks are synchronized.

In [12], the authors discussed collaborative signal processing techniques to detect, classify, and track multiple targets. The authors assumed multiple target detection at each sensor separated either in space or time, i.e., two targets are either separated by some distance or appear in two different time durations. The work described in [13, 2, 4, 6, 7, 15, 17, 25] provides similar distributed collaborative algorithms for target localization, classification, and tracking. Our mechanism can be readily extended to track multiple targets by assuming that each target has a unique ID and the local and the global hypotheses can be created for each ID.

There are some works in the literature on mining the movement patterns associated with time intervals in object tracking WSNs (OTSN). In [21], a group moving pattern mining algorithm, a prediction-based routing algorithm, and a group data aggregation algorithm are proposed. The authors contribute in two areas: first, exploring group relationship among object moving patterns and second, an efficient prediction-based query algorithm and an efficient data aggregation algorithm for OTSN. They show by experiments that the explored group relationship and group moving pattern are adopted to predict the group location such that the amount of query network traffic is significantly reduced. The amount of update network traffic that is incurred by reporting the location of monitored objects is also greatly reduced especially when the group density of monitored objects is high. In addition, the adaptive data aggregation range improves the prediction hit rate. In [22], the authors proposed a novel data mining algorithm named TMP-Mine with a special data structure named TMP-Tree for efficiently discovering the temporal movement patterns of objects. They proposed novel location prediction strategies that utilize the discovered temporal movement patterns to reduce the prediction errors. Through empirical evaluation on various simulation conditions and real dataset they also show that TMP-Mine and the proposed prediction strategies deliver excellent performance in terms of scalability, accuracy and energy efficiency. The work in [16] proposed a heterogeneous tracking model (referred to as HTM), to efficiently mine object moving patterns and track objects. Specifically, they use a variable memory Markov model to exploit the dependencies among object movements. Furthermore, due to hierarchical nature of HTM, multi-resolution object moving patterns are provided. The proposed HTM is able to accurately predict the movements of objects and thus to reduce the energy consumption for object tracking. The work presented in this paper is closest to the work in [21, 22, 16] where we propose data mining algorithm to discover the temporal movement patterns of object in WSNs.

In the context of these related works, we should emphasize that our attention is primarily focused on mining the movement patterns associated with time intervals in object tracking. We analyze these sensor stored data and extract the temporal movement patterns of object after sensor nodes detection, and capture and store timestamped coordinate information about the moving object.

3 PROBLEM FORMULATION TERMS

We consider sensor nodes to be spread randomly across a geographical area and collaborate among themselves to form a random WSN. The assumptions about the WSN are the following:

1. All sensors have the same characteristics.
2. Sensors are randomly distributed across the area of interest.

3. All sensors have the capability to capture information about any moving object in their sensing range. The information includes the approximate x, y coordinates, and the timestamp. This assumption followed based on the system model proposed by Chin-Lung [24]. The effect of sampling rate and the object velocity depends on the underlying algorithm which estimates object location. The dataset used in our algorithm consists of x and y coordinates of the event point, along with the timestamp. Each sensor may have a number of such data points recorded in its local memory.
4. Time synchronization is critical for distributed systems, and it is particularly important for WSNs that are used for gathering data corresponding to an event. Without a global agreement on time, data from different sensors cannot be matched. In our algorithm, we implicitly assume that the node clocks are synchronized.

No specific assumptions are made about the target movement pattern. However, we assume that the targets originate outside the sensing region and then move inside. Also, the aggregated data are reported to the end user.

3.1 Local Hypothesis (LH)

We define the Local Hypothesis as a set of three or more points, satisfying the following criteria:

1. Taking sets of three points or more, they should lie on the same line in the same direction.
2. The points in the LH should be in ascending timestamp manner.

The angle between the two points p_1 and p_2 can be computed by the following equation:

$$\text{Angle}(p_1, p_2) = \arctan[(\text{Ycord}(p_2) - \text{Ycord}(p_1))/(\text{Xcord}(p_2) - \text{Xcord}(p_1))] \tag{1}$$

LH is considered as a straight line, starting at the first point (FP), and ending at the last point (LP) with a specific angle. For that, we consider the structure of LH as $LH = (FP, LP, \text{angle})$, [10, 11].

3.2 Global Hypothesis (GH)

A GH is considered as a structure that contains a set of LHs. GH is formed when different LHs are merged in an ascending manner according to their timestamps. We keep track of GH direction by updating the *PointChangeList*, which includes the points at which the GH changes its direction. The GH starts at the first point of the first attached LH and ends at the last point of the last attached LH. We define the length of GH to be the number of LHs considered during forming GH. We define the GH structure as GH = (Start Point (SP), ⟨PointChangeList⟩, End Point (EP)) [10, 11].

4 ALGORITHM OUTLINES

In this section, we outline our proposed algorithm to discover global temporal patterns. First the useful sets of points (local hypotheses) are extracted from the local points stored at the sensor nodes, then from these sets of points we extract the most useful points (Global hypotheses). Second, we elect a set of sensor nodes to work as CHs; these CHs form a backbone of the WSN. Summarized GHs are aggregated over the backbone till the end user gets the global summarized GH. The outline of the whole mechanism is as follows:

4.1 Step 1: Forming the Local Hypotheses

When an object moves in the range of a sensor node, it records details of the object and arranges them in ascending order of their timestamps. We determine the angles between the p_i (i=1), and p_{i+1}, and between p_{i+1}, and p_{i+2}. If the computed angles are not equal, we skip the first point of the three points, and start from the second one. Otherwise the LH is established by setting up the first point to FP, the third point to LP, and the angle between the second and the third points to LH angle. Further points may be added to the current LH, by taking the next point p_j ($j = i + 3, \dots, N$), where N is the number of points in the database. If (p_{j-1}, p_j) angle is equal to the LH angle, we update LP to p_j.

The procedure to generate the local hypotheses LHs is as follows:

1. Sort your data in ascending order as per their timestamps,
2. Let $i = 1$,
3. While ($i < N$)
   (a) Compute the angles between p_i, p_{i+1}, and p_{i+1}, p_{i+2},
   (b) If the angles between the three points are the same,
      i Establish LH by setting up p_i to FP, and p_{i+2} to LP of the LH,
      ii For every next point p_j ($j = i + 3$ to N),
         A Compute the angle between p_j, and last added point to the current LH (p_{j-1}),
         B If the computed angle equals to the previous one in 3(a),
            - Update LP to p_j,
         C Else start creating a new LH by setting $i = j$, and go to 3,
      iii End for
   (c) Else $i = i + 1$ and go to 3,
4. End while.
5. End Procedure.

4.2 Step 2: Computing the Global Hypothesis

In this step, each sensor extracts the most important points among the set of LHs. Each sensor computes its GH by merging its LHs, using the following procedure:

1. Arrange your LHs in ascending manner as per their last point timestamps,
2. Set $PointChangeList$ to Φ,
3. Set the FP of the first LH as the start point of the GH,
4. For every next LH,
   (a) If the angle of the current LH is not equal to the angle of the previous LH,
      i Add the FP and the angle of the current LH into $PointChangeList$ of the GH,
5. End for
6. Set the LP of the last taken LH as endpoint of the GH,
7. End procedure.

4.3 Step 3: CHs Election and Backbone Creation

In this step, every sensor node broadcasts the endpoint timestamp of its GH to its neighboring nodes. The sensor node that possesses the maximum endpoint timestamp considers itself as root node (S_{Root}) and volunteers itself as the CH and sets its $RootId$, and $ParentId$. S_{Root} initiates the WSN clustering process by finding the sensor (say S_j) with minimum timestamp from its members and urges it to be a member of its cluster. S_j updates its $RootId$ to the supporter $RootId$, and its $ParentId$ to the supporter Id. Then, S_j volunteers itself as a CH and forms a new cluster with neighboring nodes that are not members in any other cluster.

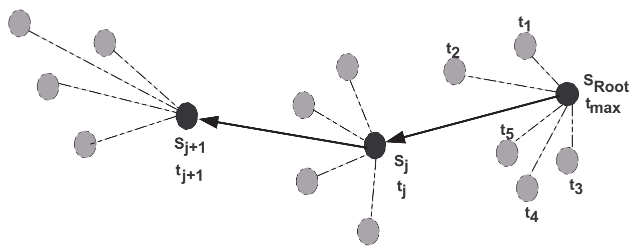


Fig. 1. Clustering processes

Figure 1 shows how the election process occurs in our mechanism; the black nodes are the elected CHs while the gray nodes are the cluster members. S_{Root} possesses the maximum timestamp in the WSN for that it initiates the clustering process and forms a cluster as shown by gray nodes around S_{Root}. S_{Root} finds that t_j is the minimum timestamp from the received timestamps (t_1, t_2, t_3, t_4, t_5, t_j) and sends message to S_j to form a new cluster. S_j sets its *ParentId* to the Id of S_{Root} and its *RootId* to the *RootId* of S_{Root} and volunteers itself as a CH. In a similar way, S_j supports S_{j+1} that has the minimum timestamp received by S_j.

After broadcasting the GH endpoint timestamp of every sensor, the election process of CHs can be done by the following procedure at every sensor S_i:

1. arrange the received GHs according to their endpoint timestamps in ascending manner,
2. if you have the largest timestamp in the WSN
   (a) set yourself as S_{Root} node for the WSN,
   (b) set your RootId and ParentId to your Id,
   (c) volunteer yourself as CH by broadcasting declaration message,
   (d) find the sensor node that reports minimum timestamp (say S_j),
   (e) initiate clustering process by transmitting support message to S_j.
3. end if
4. if you receive support message,
   (a) set your ParentId to the Supporter Id,
   (b) find the sensor node that reported minimum timestamp (say S_j),
   (c) if your timestamp is greater than the timestamp of S_j
      i volunteer yourself as CH by broadcasting declaration message,
      ii transmit support message to S_j.
5. end if
6. if you are not cluster member node AND a declaration message has been received, join the cluster by transmitting confirmation message to the CH.
7. end if
8. end Procedure

By the end of step 3, the skeleton of the WSN is formed and the backbone is the volunteered CHs. Global summarized GH is formed by merging all the summarized GHs at the CHs. The last CH at the backbone reports the global summarized GH to the end user.

4.4 Step 4: Computing Summarized GH and Reporting the Results

In this step, we aggregate the generated GHs to get the global summarized GH. This will be done in two levels, the first level will be between the sensor nodes and their CH, while the second one will be among the CHs.

At the cluster members level each CH forms its summarized GH by merging the received GHs.

At the CHs level the generated GHs at each CH will be aggregated till we obtain the global summarized GH. At this level, lower CH which has minimum $GH.EP$ timestamp will send its GH to the next cluster head. The next CH merges the received GH into its GH to form a new GH and then sends this new GH to its next CH and so on till we get the global summarized GH at the last CH. The last CH will report the GH to the end user.

The following procedure shows how the merging operation of two GHs can be executed at any sensor node.

1. Set the minimum SP (SP with minimum timestamp) of the two GHs, to the SP of the new GH.
2. Set the union of PointChangeLists of the two GHs to the PointChangeList of the new GH, arrange them according to their timestamps.
3. Set the maximum EP (EP with maximum timestamp) of the two GHs endpoints, to the EP of the new GH.
4. End Procedure.

For more clarification consider the following network.

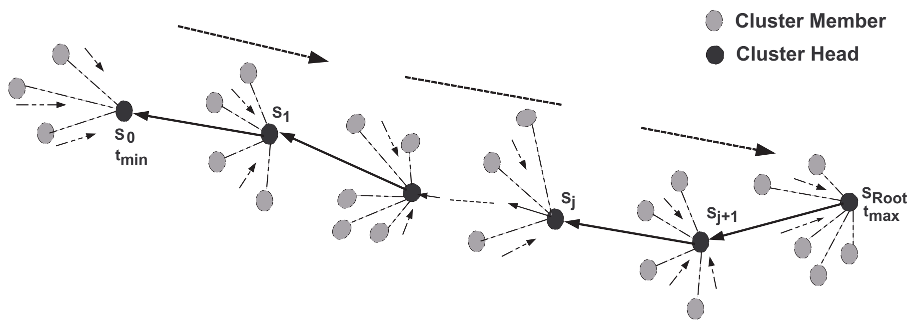


Fig. 2. Network Backbone and aggregation levels

In Figure 2, the gray nodes represent cluster members while the black nodes represent CHs. Each black node aggregates the GHs from its gray nodes to form the summarized GH. The next level of aggregation process begins from node S_0 that has $t_{\min}$ toward S_{Root} that has $t_{\max}$, where at each CH except the first one a new summarized GH is formed by merging the received GH with its GH. This merging process will be continued till S_{Root} merges its GH with the received GH from S_{j+1}, and reports the result to the end user.

5 SIMULATION RESULTS

We analyze the performance of our algorithm via simulation using NS2. The WSN model for simulation consists of randomly placed sensor nodes in a constant square area ($300 \times 300\,\mathrm{m}^2$). In our simulation, we focus on measuring the energy consumption as performance between our approach and the direct communication architecture (DC).

In order to measure the energy dissipation of sensor nodes, we use the same energy parameters and radio model as discussed in [8], wherein energy consumption is mainly divided into two parts: receiving and transmitting messages. The transmission energy consumption needs additional energy to amplify the signal depending on the distance to the destination. Thus, to transmit a k-bit message a distance d, the radio expends will be,

$$E_{Tx}(k,d) = \begin{cases} kE_{\mathrm{elec}} + k\epsilon_{\mathrm{fs}}d^2 & d < (\epsilon_{\mathrm{fs}}/\epsilon_{\mathrm{mp}}) \\ kE_{\mathrm{elec}} + k\epsilon_{\mathrm{mp}}d^4 & d \geq (\epsilon_{\mathrm{fs}}/\epsilon_{\mathrm{mp}}) \end{cases} \tag{2}$$

and to receive this message, the radio expends will be

$$E_{Rx}(k) = k \cdot E_{elec}. \tag{3}$$

Simulated model parameters are set as: $E_{\mathrm{elec}} = 50\,\mathrm{nJ/bit}$, $\epsilon_{\mathrm{fs}} = 10\,\mathrm{pJ/bit/m^2}$, $\epsilon_{\mathrm{mp}} = \frac{13}{10\,000}\,\mathrm{pJ/bit/m^4}$ and the initial energy per node $= 2\,\mathrm{J}$. We study the effect of the number of sensor nodes and the transmission radius parameters on the WSN energy dissipation.

The first part of our simulation shows the effect of increasing sensing range on the WSN energy dissipation of our approach and *DC* approach. We run the test with 300 sensor nodes, and increase the sensing range from 30 m to 60 m in increment of 5 m. Figure 3 shows that the total energy consumption in our approach is much less than the consumption energy of the *DC* approach. This could be due to the increase of the probability of neighbors of each sensor node and to the decrease of the number of CHs.

In the second part of our simulation, we measure the effect of number of sensor nodes on the energy dissipation of nodes. We fix the sensing range of the sensor nodes at 50 m and the communication range to be approximately twice of the sensing range. The number of sensor nodes varies from 100 to 500 in increment of 100 sensor nodes. Figure 4 shows that the energy consumption in our approach is less than that of the *DC*. This could be due to the fact that in our approach each node maximizes its local computations and minimizes the exchanged information.

The simulation results show that the performance of our approach performs well and can achieve remarkably low energy consumption. In addition, our results show that the algorithm works effectively irrespective of the number of sensor nodes, i.e. our approach is scalable for increasing the number of sensor nodes.

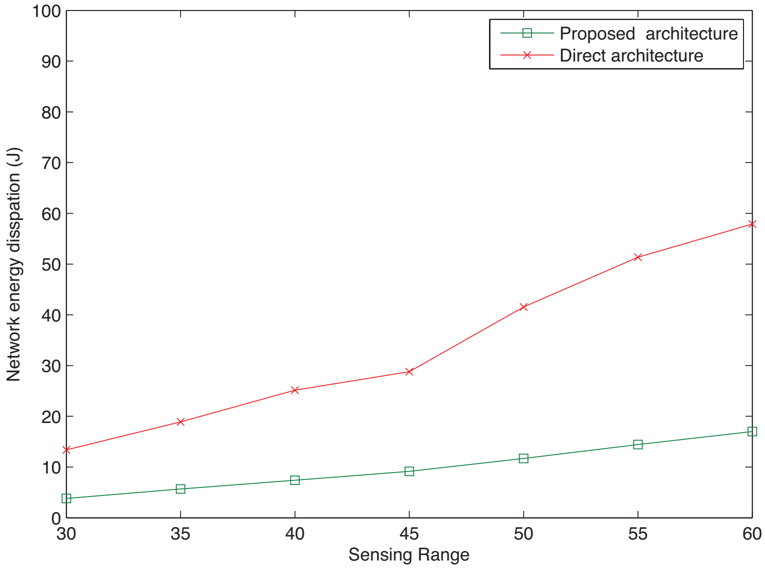


Fig. 3. Impact of sensing range

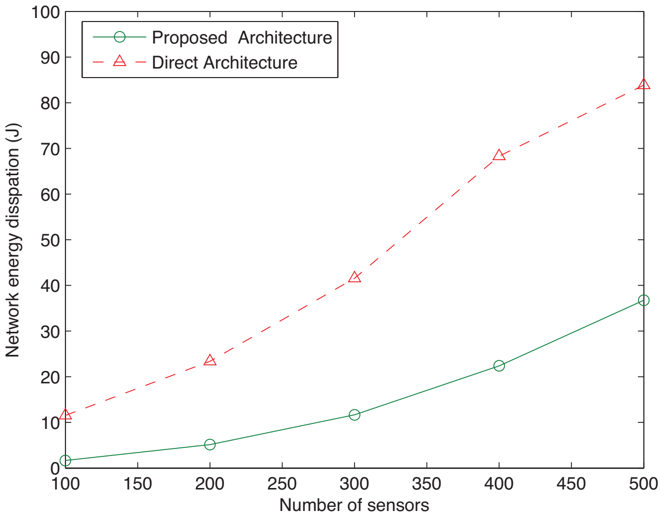


Fig. 4. Performance for different number of sensors

6 COMMUNICATION REQUIREMENTS

In this section, we present the communication requirements of our proposed mechanism. Since the computation cost at each sensor node is small as compared to the communication cost of two sensors, the communication requirement comes from the number of exchanged messages. In our algorithm, there are four steps to be preformed till the creation of global summarized GH. We analyze each one of these four steps in terms of the number of exchanged messages as follows:

1. **Forming LHs and GHs:** The computations of forming LHs and GHs are locally performed, so that no exchanged messages are needed.
2. **CHs election:** If we assume that m is the total number of sensors that store object data, and l is the number of elected CHs, the exchanged messages for this step will be as follows:
   - m messages are required to the GHs,
   - l messages are required for the volunteered CHs,
   - $m - l$ confirmation messages from neighbors (cluster members) to their CH, and
   - l support messages from prior CH to the next one.
3. **Computing summarized GH and reporting the results:** The exchanged messages for this step will be as follows:
   - l messages are required for communication at the CH level till the global summarized GH is obtained.

Therefore, the total number of messages for our algorithm will be:

$$\text{Total Number of Exchanged Messages} = 2m + 2l \in O(m). \tag{4}$$

7 CONCLUSION

In this paper, we have proposed an algorithm for target tracking by discovering temporal patterns from a set of distributed sensor databases containing temporal data. The concept of maximizing the computations at the local site and minimizing the exchange of information between the sensor nodes helps reduce the load on the overall WSN; this formed the crux of our algorithm. In our work, we considered the problem of mining temporal data in distributed datasets. We worked with sensor nodes which are capable of capturing and storing approximate coordinate information about a moving object. These sensor nodes are placed randomly and our algorithm is used to discover the nonlinear trajectory of the moving object. We considered this equivalence to mining of global spatiotemporal patterns from geographically distributed datasets. Since there is only one database scanning required in the beginning, the complexity of our algorithm is drastically reduced.

Acknowledgment

The authors would like to thank Prof. Dharma P. Agrawal for helpful comments that improved overall quality of this paper. We also thank the referees for their comments.

REFERENCES

[1] Amundson, E. I.—Kushwaha, M.—Kusy, B.—Volgyesi, P.—Simon, G.—Koutsoukos, X.—Ledeczi, A.: Time Synchronization for Multi-Modal Target Tracking in Heterogeneous Sensor Networks. Workshop on Networked Distributed Systems for Intelligent Sensing and Control, June 2007.

[2] Aslam, J.—Butler, Z.—Constantin, F.—Crespi, V.—Cybenko, G.—Rus, D.: Tracking a Moving Object with a Binary Sensor Network. The First ACM Conference on Embedded Networked Sensor Systems (Sensys03), Los Angeles, CA (USA), November 2003.

[3] Bandyopadhyay, C.—Giannella, C.—Maulik, U.—Kargupta, H.—Liu, K.—Datta, S.: Clustering Distributed Data Streams in Peer-To-Peer Environments. Information Sciences, Vol. 176, 2006, No. 14, pp. 1952–1985.

[4] Fang, Q.—Zhao, F.—Guibas, L.: Counting Targets: Building and Managing Aggregates in Wireless Sensor Networks. Xerox Palo Alto Research Center (PARC) Technical Report, June 2002, pp. 202–298.

[5] Guha, S.—Murty, R.—Sirer, E. G.: Sextant: A Unified Node and Event Localization Framework Using Non-Convex Constraints. In Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Urbana-Champaign, IL, May 2005, pp. 205–216.

[6] Hao, Q.—Brady, D.-J.—Guenther, B. D.—Burchett, J.—Shankar, M.—Feller, S.: Human Tracking With Wireless Distributed Pyroelectric Sensors. IEEE Sensors Journal, Vol. 6, 2006, No. 6, pp. 1683–1696.

[7] Hatzinakos, D.—Liang, S.: A Cross Layer Architecture of Wireless Sensor Networks for Target Tracking. ACM Transactions on Networking, Vol. 15, 2007, No. 1, pp. 145–158.

[8] Heinzelman, W.—Chandrakasan, A.—Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Trans. Wireless Comm., Vol. 1, 2002, No. 4, pp. 660–670.

[9] Jeong, J.—Hwang, T.—He, T.—Du, D.: MCTA: Target Tracking Algorithm Based on Minimal Contour in Wireless Sensor Networks – INFOCOM 2007. 26^{th} IEEE International Conference on Computer Communications, May 2007, pp. 2371–2375.

[10] Khedr, A. M.—Osamy, W.: Tracking Mobile Targets Using Random Sensor Networks. The Arabian Journal for Science and Engineering, Vol. 32, No. 2B, 2007, pp. 301–315.

[11] Khedr, A. M.: New Mechanism for Tracking a Mobile Target Using Grid Sensor Networks. Computing and Informatics, Vol. 28, 2008, pp. 1001–1021.

[12] Li, D.—Wong, K.—Hu, Y.—Sayeed, A.: Detection, Classification, and Tracking of Targets. IEEE Signal Processing Magazine, Vol. 19, 2002, pp. 17–29.

[13] Liu, J.—Reich, J.—Cheung, P.—Zhao, F.: Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications. 2nd Workshop on Information Processing in Sensor Networks (IPSN 03), Palo Alto, California, April 2003.

[14] Park, S. Y.—Ahn, H. S.—Yu, W.: A Simple Object Tracking System Using ITDOA without Time Synchronization. The 9th International Conference on Advanced Communication Technology, Gangwon-Do, 12–14 February 2007, Vol. 3, pp. 2026–2028.

[15] Pattem, S.—Poduri, S.—Krishnamachari, B.: Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks. 2nd Workshop on Information Processing in Sensor Networks (IPSN 03), Palo Alto, California, April 2003.

[16] Peng, W.-C.—Ko, Y.-Z.—Lee, W. C.: On Mining Moving Patterns for Object Tracking Sensor Networks. 7th International Conference on Mobile Data Management (MDM 2006), Nara (Japan), May 9–13, pp. 41–49.

[17] Ramanathan, P.: Location-centric Approach for Collaborative Target Detection, Classification, and Tracking. Proceedings of IEEE CAS Workshop on Wireless Communication and Networking, Pasadena, California, September 2002.

[18] Ren, Q.—Liang, Q.: Energy and Quality Aware Query Processing in Wireless Sensor Database Systems. Information Sciences, Vol. 177, 2007, No. 10, pp. 2188–2205.

[19] Simon, G.—Maroti, M.—Ledeczi, A.—Balogh, G.—Kusy, B.—Nadas, A.—Pap, G.—Sallai, J.—Frampton, K.: Sensor Network-Based Countersniper System. In: SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, ACM Press, New York 2004, pp. 1–12.

[20] Sundararaman, B.—Buy, U.—Kshemkalyani, A. D.: Clock Synchronization for Wireless Sensor Networks: A Survey. Ad Hoc Networks, Vol. 3, 2005, No. 3, pp. 281–323.

[21] Tsai, H. P.—Yang, D. N.—Peng, W. C.–Chen, M. S.: Exploring Group Moving Pattern for an Energy-Constrained Object Tracking Sensor Network. Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007), Nanjing, China, May 22–25, 2007.

[22] Tseng, V. S.—Lin, K. W.: Energy Efficient Strategies for Object Tracking in Sensor Networks: A Data Mining Approach. J. Syst. Softw. 80, 10 Oct. 07, pp. 1678–1698.

[23] Walchli, M.—Scheidegger, M.—Braun, T.: Intensity-Based Event Localization in Wireless Sensor Networks. In: Proceedings of IFIP Third Annual Conference on Wireless On Demand Network Systems and Services (WONS '06), Les Menuires, France, January 2006.

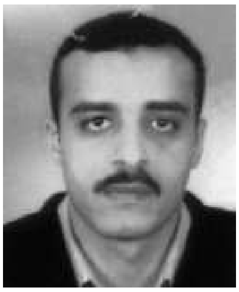
[24] Yang, C. L.—Bagchi, S.—Chappell, W. J.: Location Tracking with Directional Antennas in Wireless Sensor Networks. IEEE MTT-S International Microwave Symposium (IMS), Long Beach, California, June 11–17, 2005.

[25] Yeow, W. L.—Tham, C. K.—Wong, W. C.: Energy Efficient Multiple Target Tracking in Wireless Sensor Networks. IEEE Transactions on Vehicular Technology, Vol. 56, 2007, No. 2, pp. 918–928.

[26] Zou, Y.—Chakrabarty, K.: Sensor Deployment and Target Localization Based on Virtual Forces. In: Proceedings of IEEE INFOCOM, 2003.

[27] Zou, Y.—Chakrabarty, K.: Sensor Deployment and Target Localization in Distributed Sensor Networks. IEEE Trans. on Embedded Computing Systems, Vol. 3, 2004, No. 1, pp. 61–91.

Ahmed M. Khedr received his B. Sc. degree in mathematics in June 1989 and the M. Sc. degree in optimal control in July 1995, both from Zagazig University, Egypt. In July 1999 he received his M. Sc. and in March 2003 he received his Ph. D. degree, both in computer science and engineering, from University of Cincinnati, Ohio, USA. From March 2003 to January 2004, he was a Research Assistant Professor at ECECS Department, University of Cincinnati, USA. From January 2004 to May 2009, he worked as Assistant Professor at Zagazig University, Egypt, and from June 2009 till now he is working as Associate Professor at the Department of Computer Sciences, College of Computers and Information Systems, Taif University, KSA. In June 2009, he was awarded the State Prize of Distinction in Advanced Technology. He has co-authored 32 works in journals and conferences related to optimal control, wireless sensor networks, decomposable algorithms, and bioinformatics.



Walid Osamy received his B. Sc. degree in 2000 and his M. Sc. degree in 2006, both from the Department of Mathematics, Faculty of Science, Zagazig University, Egypt. He has been involved in the projects on Network Infrastructure, telecom, and Managment information systems at Communication Information Technology Center (CITC), Zagazig University. He is currently a Ph. D. student at the Faculty of science, Zagazig University. His research interests include computer graphics, computing, and wireless sensor networks.