Hierarchical Spatial Clustering in Multi-hop Wireless Sensor Networks

Zhidan Liu*†, Wei Xing✉ *†, Bo Zeng*†, Yongchao Wang†, Dongming Lu*†
*College of Computer Science, Zhejiang University, Hangzhou, 310027, P. R. China
†Cyrus Tang Center for Sensor Materials and Applications, Zhejiang University, Hangzhou, 310027, P. R. China
Email: {danielliu,wxing,zb9364,ychwang,ldm}@zju.edu.cn

***Abstract*—In wireless sensor networks, the resource-limited sensor nodes collect and transmit sensing readings to Sink node collaboratively in multi-hop manner to conserve energy. For more energy-efficient data collection, we consider the problem of spatial clustering, which aims to group the strong correlated sensor nodes into the same cluster for rotatively reporting representative data later, and propose a hierarchical spatial clustering algorithm named as HSC. Specifically, with similarity measure on both magnitude and trend of sensing readings, HSC groups the similar sensor nodes in distributed and hierarchical manner by exploiting a pre-built data collection tree, which dispenses HSC from the extra requirements, such as global network topology and strict time synchronization, during clustering. Extensive simulation results show that HSC performs superiorly on clustering quality when compared with the alternative algorithms. Furthermore, approximate data collection scheme combined with HSC can reduce much more communication overhead while incurring modest data error than with other algorithms. In general, HSC possesses comprehensive advantage on the aspects of both clustering quality and approximation performance.**

I. Introduction

Recent years, wireless sensor networks (WSNs) have been witnessed the wide utilization in many applications, such as environmental monitoring, habitat monitoring and structural monitoring [1]. These WSNs typically consist of a huge number of sensor nodes. Due to significant constraints in the amount of available resources, in especial the energy budget, these tiny sensor nodes usually form a multi-hop communication network and transmit sensing readings to Sink cooperatively via a pre-built data collection tree.

As wireless communication consumes the most energy among all the activities of a sensor node, reducing the communication overhead has been the uppermost principle of prolonging the lifetime of WSNs. By exploiting the tradeoff between data quality and energy consumption, approximate data collection could be a rational choice for long term data-driven applications as some argue that approximate data could be still precise enough for some WSN-based applications, which can tolerate certain accuracy loss of sensing readings to perform data analysis or decision making [2]. As ubiquitous spatial correlation among nearby sensor nodes, some representative sensor nodes can be scheduled to transmit their readings to approximate the sensing data of their spatial correlated sensor nodes accordingly [3]. Spatial clustering, which combines the clustering technique and spatial correlation, becomes to be an effective approach to find these similar node sets, sensor nodes in which could represent each other during data collection. Generally speaking, the two core problems of spatial clustering are how to evaluate the similarity between sensor nodes and how to group those nodes with similar data distribution into the same cluster in an energy-efficient manner. Previous work mainly measures the magnitude similarity with raw readings, but overlooks the importance of trend similarity [3] [4] [5] [6]. Yet some work does exactly the opposite [7]. We argue that both magnitude similarity and trend similarity hold the equivalent importance. Besides, due to the *ad hoc* nature and resource constraints, distributed spatial clustering, which takes advantages of the structure of WSNs, would be a better choice.

As typical time series data, readings of sensor nodes can be modeled as AutoRegressive (*AR*) model to forecast the data in the near future [8]. In other words, an appropriate *AR* model could capture the trend of sensor data in an easy way. In this paper, based on the *AR* model and the pre-built data collection tree, we propose the **H**ierarchical **S**patial **C**lustering (**HSC**) algorithm to group similar sensor nodes. Unlike previous work with consideration of magnitude similarity alone, we define a novel method to measure the nodes similarity by exploiting the coefficients of their *AR* models to take into account the trend similarity too. In addition, by effectively exploiting the data collection tree, our algorithm can complete the clustering task gracefully without other requirements, e.g., global topology information or strict time synchronization. Compared with alternative algorithms, simulation results show that our algorithm offers advantages on both the clustering quality and the approximation performance.

The contributions of this paper are summarized as follows:

- We have designed a novel sensor nodes similarity measure method, which jointly takes magnitude similarity and trend similarity of sensor readings into considerations.
- With our similarity measure method, we have proposed a hierarchical spatial clustering algorithm which exploits the pre-existing structure of multi-hop WSNs effectively.

The rest of this paper is structured as follows: Section II briefly discusses the related work. System model is described in Section III, and the details of HSC algorithm are elaborated in Section IV. Performance evaluation is presented in Section V, and finally with conclusion in Section VI.

✉Corresponding author.

978-1-4673-2815-9/13/$31.00 ©2013 IEEE

II. Related Work

In this section, we will present the related works of spatial clustering, as well as related works of *AR* model in WSNs.

A. Spatial Clustering

In this sub-section, we concentrate on most related work on spatial clustering for approximate data collection. A well-reviewed survey of clustering algorithms is referred to [9].

Spatial clustering captures the underlying spatio-temporal pattern in WSNs, and two relevant problems are how to measure the similarity between nodes and how to group these nodes with similar readings in an energy-efficient manner. Some previous works measure the similarity with raw readings based on *Manhattan distance* metric, mostly only on the magnitude similarity, such as DCglobal [4], DClocal [5], IADSC [10], DACA [11] and DSCC [12]. Besides considering the magnitude similarity of raw readings, EEDC [3] also takes into account the trend similarity during nodes similarity measure. EEDC models this spatial clustering problem as clique-covering problem which is proved NP-hard, and solves it with a greedy algorithm in a centralized fashion. Obviously, EEDC will lead to huge energy consumption as it need to collect enough raw reading from all sensor nodes. Even worse, spatial clustering with similarity measure sole on raw readings may form "unsteady" spatial clusters.

ELink [7] emphasizes that spatial clustering should be performed on the underlying trend rather than on the raw time series readings. In ELink, each node constructs *AR* model to capture the structure of time series readings, and computes its feature in the metric space of weighted *Euclidean distance* with *AR* model coefficients. Based on these features, ELink models the clustering problem as δ-clustering, where feature distance between any two nodes in the same cluster is less than δ. Recursively, some well distributed nodes are chose as sentinel nodes to expand their clusters until they are δ-compact in distributed manner. Though taking the underlying trend of time series readings into consideration, ELink overlooks the baseline value of each node. In other words, sensor nodes in the same cluster could have the similar variation tendency, but their real sensing values may differ greatly. Furthermore, the selection of well-distributed sentinel nodes requires a priori information of global network topology, which will weaken the applicability of ELink in practice. SAF [6] also exploits the *AR* model when clustering. However, SAF still only depends on the magnitude similarity of sensor nodes but leaves out the trend similarity. As mentioned, our algorithm, i.e., HSC, takes both magnitude and trend similarity into considerations, and group the similar nodes in distributed and hierarchical manner based on the pre-built data collection tree of WSNs.

B. AutoRegressive Model

AR model is a linear regression, and is used to capture the dependency of current value and its latest historical values in time series data analysis. Obviously, sensor readings are typical time series data, and the *AR* model could be adopted for data forecasting in WSNs. Though simple and light-weight, the *AR* model offers excellent accuracy in WSNs for monitoring applications [8]. An *AR* model with order p, which is the number of lagged values in a linear regression, is usually denoted as $AR(p)$, and expressed as

$$x_t = c + \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon, \tag{1}$$

where $\vec{\alpha} = \{\alpha_1, ..., \alpha_p\}$ are the coefficients of *AR* model, c is a constant but always omitted for simplicity, and ϵ is the *White Noise*. The calibration of *AR* model is quite simple, and various methods can be used for parameter estimation of *AR* model. More details of *AR* model and time series data analysis are referred to [13].

Due to low computation overhead and memory requirements yet outstanding performance in data forecasting, *AR* model has been widely used for approximate data collection in WSNs. PAQ [8] adopts *AR* model to approximate the values of sensor nodes. With *AR* model built locally, each sensor node transmits its model to Sink, which uses these models to predict values of nodes without direct communication. Besides, PAQ proposes a dynamic *AR* model maintaining strategy to cope with the changing environment. With the *dual-prediction* based data collection idea in mind, quite a few similar work [2] [3] building on *AR* model has been presented for prolonging the network lifetime of WSNs by keeping sensor nodes from transmitting redundant data.

III. System Model

In this paper, we consider a sensor network consisting of a collection $S = \{s_1, s_2, ..., s_N\}$ of N sensor nodes and one Sink node. All sensor nodes are uniformly and randomly distributed in a sensing field of size $L \times L$. The Sink node is located outside of the sensing field but not far away. All sensor nodes have the identical communication radius R, and communicate with distant nodes hop by hop. Periodically, sensor nodes generate environmental sensing readings which evolve over time, and transmit these readings to Sink via *data collection tree* (DCT). According to the DCT, each sensor node learns its unique parent node, dct_parent, and maintains a list of its direct children nodes, $dct\_children$.

Furthermore, each sensor node maintains a queue to store the most recent W sensing readings and keeps tracking of the average value μ of the W data. By exploiting these data in queue, each sensor node can learn an *AR* model using least-square method. Similar to PAQ [8], we also adopt a narrow prediction window, such as $p = 3$, to neglect the impact of non-stationary physical environment, and to make the *AR* model simple and light-weight enough for resource-limited sensor nodes as well. To cope with dynamic environment and keep accuracy of *AR* models, we adopt the model monitoring algorithm of PAQ to maintain a dynamic local *AR* model for each node, namely, sensor node will re-learn its *AR* model using the latest W data in queue if the number of times the prediction error beyond error threshold θ within consecutive Λ epochs exceeds the pre-defined threshold ν.

IV. The Proposed Spatial Clustering Algorithm

A. Preliminary

Before presenting the details of HSC algorithm, we first introduce our method to measure the similarity between any two sensor nodes. As been proved in our previous work [14], *AR* model shows its powerful capability to capture the trend of time series data in data collection of sensor networks. In addition, average value μ of data queue maintained at each sensor node is an ideal baseline to represent the reading magnitude of the monitored region. Consequently, it is feasible and reasonable to use the *AR* models and the average values μ of any two sensor nodes to measure their similarity. As typical linear systems, correlation between two *AR* modes can be well measured with *Pearson Correlation Coefficient*. Formally, assume two nodes s_i and s_j with their *AR*(3) model coefficients $\vec{X} = \{x_1, x_2, x_3\}$ and $\vec{Y} = \{y_1, y_2, y_3\}$ respectively, *trend similarity* between s_i and s_j can be calculated as

$$\rho_{s_i,s_j} = \frac{\mathrm{cov}(\vec{X}, \vec{Y})}{\sigma_{\vec{X}}\sigma_{\vec{Y}}} = \frac{E((\vec{X} - \mu_{\vec{X}})(\vec{Y} - \mu_{\vec{Y}}))}{\sigma_{\vec{X}}\sigma_{\vec{Y}}}. \quad (2)$$

In addition, we adopt *Manhattan distance* to measure the magnitude similarity between their baseline values, namely that if their *magnitude similarity* $M_{s_i,s_j} = \left|\mu_{s_i} - \mu_{s_j}\right| \leq \frac{\varepsilon}{2}$, then these two nodes are considered magnitude similar, where ε is a user-defined parameter to bound the maximum difference between sensing readings of any two sensor nodes in the same cluster. Hence, we could have the following definition to judge whether two sensor nodes are similar.

Definition 4.1: *similar nodes*. two sensor nodes, s_i and s_j, are similar nodes and can be grouped into the same cluster if their trend similarity $\rho_{s_i,s_j} \geq cth$ and magnitude similarity $M_{s_i,s_j} \leq \frac{\varepsilon}{2}$, where both cth and ε are user-defined application-dependent parameters to guide the spatial clustering.

With the definition of similar nodes, we could have a formal definition about the problem of spatial clustering.

Definition 4.2: *spatial clustering problem*. For a *multi-hop* WSN with a collection $S = \{s_1, s_2, ..., s_N\}$ of N nodes, the whole sensor network can be partitioned into definite node sets $C = \{C_1, C_2, ..., C_y\}$, where $\bigcup_{a=1}^{y} C_a = S$. For $\forall s_i, s_j \in C_a$, s_i and s_j are similar nodes, i.e., $\rho_{s_i,s_j} \geq cth$ and $M_{s_i,s_j} \leq \frac{\varepsilon}{2}$. Spatial clustering aims to group similar nodes into the same cluster, and meanwhile to obtain $\min y$.

Actually, there are two objectives for spatial clustering. One is to ensure all sensor nodes in the same cluster are remarkably similar, and the other is to minimize the number of these kind of clusters. The two objectives together guarantee the data accuracy and energy efficiency of approximate data collection. Unlike some clustering algorithms which perform clustering first and then build the DCT, we argue that spatial clustering should not break the pre-built DCT as it may be built with some optimal algorithms, e.g. [15], which build DCT better than those built based on clusters. Due to these considerations, spatial clustering is quite a challenging task, especially in the *multi-hop* scenario.

B. The HSC Algorithm

Generally, the HSC algorithm includes two major phases, namely, the local *AR* model learning phase and the hierarchical spatial clustering phase.

1) The local AR model learning phase: To avoid transmitting abundant raw sensing readings to Sink to build probabilistic model for each node, we prefer to learn and maintain the *AR* model locally at each sensor node. After accumulating enough data, i.e., W data to feed the queue full, each sensor node will estimate the coefficients of *AR*(3) by calculating the minimum square error between the real readings contained in the queue of that node and the predicted values via least-square regression method. Note that other parameter estimation methods for *AR* model, i.e., maximum likelihood, could be used, but least-square regression could be a more proper method as it is simple enough to avoid complex computation.

2) The hierarchical spatial clustering phase: Once a sensor node completes the *AR* model learning phase, it then transmits the coefficient vector $\vec{\alpha}$ and the average value μ to Sink via DCT. During the parameters transmission, each intermediate node backups the average values of its direct children nodes in DCT, and stores these values in buffer Γ. With all model coefficients transmitted to Sink, it facilitates the combination of our clustering algorithm and the dual-prediction based approximate data collection scheme, which will be illustrated in section V. In principle, HSC is initiated by the Sink and it extends the clustering top-down along with the DCT. In HSC spatial clustering, there are three primary roles for sensor nodes: trigger node, sentinel node and expanding node.

- **Trigger node**: a trigger node selects the minimum number of sentinel nodes from its $dct\_children$ to cover all of its children nodes based on their average values stored in buffer Γ. Formally, a node s_i can be covered by another node s_j if they are magnitude similar, i.e, $M_{s_i,s_j} \leq \frac{\varepsilon}{2}$. To select as few sentinel nodes as possible, we adopts *median-value-first* strategy to choose sentinel nodes. First, the node whose average value is the median value among all data in buffer Γ is chosen, and then those nodes whose average values can be covered by the median value are sought out. If there are still some uncovered values, we repeat this strategy at the separated upper part and lower part un-covered values of the median value independently until all values can be covered by a sentinel node. This operation is performed among average values of nearby nodes which share similar observations, thus the *median-value-first* strategy is feasible and can terminate quickly. At last, trigger node will transmit the complete selection result to all of its $dct\_children$ nodes.
- **Sentinel node**: a sentinel node is the clusterhead of a spatial cluster. After being chosen as a sentinel node by trigger node, this node broadcasts an *Invitation* message, encoded as $\langle \vec{\alpha}, \mu, hops \rangle$ which includs the model coefficients $\vec{\alpha}$, average value μ and hop distance $hops$ from this sensor node to Sink node, to its one-hop neighbors. However, ***only*** the $dct\_children$ and sibling nodes of

this sentinel node will respond to this *Invitation* message with feedback as either *Join* message or *Reject* message, depending on whether that node is similar or dissimilar with current sentinel node.

- **Expanding node**: after receiving an *Invitation* message from sentinel node, a sensor node will first check whether it is similar with the sentinel node, if not it replies *Reject* message to that sentinel node immediately. If similar, then this sensor node becomes an expending node, and then forwards this *Invitation* message ***only*** to its children nodes in DCT to expand current spatial cluster.

Note that a sentinel node or an expanding node may switch to be trigger node if that node finds some of its $dct\_children$ are not similar with current sentinel nodes, namely, that node should select several new sentinel nodes from the un-clustered nodes to grow some new clusters. A trigger node do not finish its duty until all of its $dct\_children$ are either sentinel nodes or clustered by a sentinel node.

When Sink has received all model coefficients and average values, it becomes the first trigger node and selects several sentinel nodes from nodes in the one-hop range of Sink. With no guarantee of strict time synchronization among sensor nodes, HSC adopts the *Request-ACK* mechanism to ensure every communication during clustering is complete. Spatial clustering iterates among sensor nodes top-down along with DCT, and each node replies its dct_parent node with *Reject* message immediately or *ACK* message when it has received feedback messages, either *Reject* or *ACK*, from all of its $dct\_children$. A sensor node, no matter the role of sentinel node or expanding node, will switch to be trigger node when it finds un-covered $dct\_children$ nodes. The new trigger node will select some sentinel nodes among all of its un-clustered $dct\_children$ nodes to continue the spatial clustering.

Fig.1 shows a sample execution of HSC clustering with user-defined parameters $\varepsilon = 0.5$ and $cth = 0.9$. Box attached with each node describes the baseline value μ and correlation ρ with neighboring nodes of current node. Node s_1 is selected as a sentinel node by its dct_parent and prepares to expand its cluster with *Invitation* messages disseminated down along DCT. Node s_2 is similar with clusterhead s_1 and becomes an expanding node to forward this *Invitation* to its $dct\_children$. However, both s_4 and s_5 are dissimilar with current sentinel node, i.e., s_1. After receiving the *Reject* messages from its $dct\_children$, node s_2 turns to be trigger node and selects s_5 as the sentinel node from the un-clustered set $\{s_4, s_5\}$ with *median-value-first* strategy. Meanwhile, s_2 sends *ACK* to s_1 to affirm the cluster relation. With the assignment, s_5 broadcasts *Invitation* messages to one-hop neighbors. As sibling node, s_4 checks the message and finally accepts the invitation to expand this cluster. On the other hand, s_3 rejects the invitation from s_1 once finding that it does not meet the conditions of similar nodes with s_1. As the only un-clustered $dct\_children$ node, node s_3 is acknowledged as sentinel node by node s_1 to form another cluster. Similarly, spatial clustering is hierarchically performed among the remaining nodes. Finally, there are four clusters and nodes in each cluster form a cluster tree.

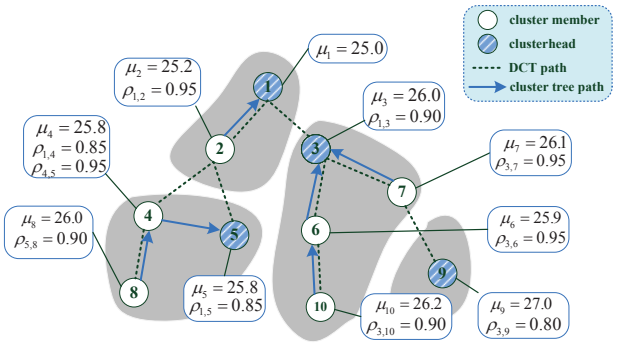

Fig. 1. A sample execution of HSC algorithm

V. Performance Evaluation

In this section, we will perform simulations with *Matlab* to first study the clustering quality of HSC. Furthermore, we study the efficiency of HSC in approximate data collection on both the energy efficiency and accuracy of collected data. Note that three alternative algorithms are implemented for contrast study in both simulation sets.

A. Simulation Setup

1) Compared algorithms: as emphasized in section II, the key factors of spatial clustering are the similarity measure and the fashion to group similar nodes into the same cluster. Therefore, we select two noteworthy previous work, namely EEDC (multi-hop version) and ELink, as the alternative algorithms. For fairness, we also design a new compared algorithm named as μELink, which is modified based on ELink to consider μ when clustering. In μELink, both feature distance and average values μ are considered in similarity measure with similar method as HSC.

2) Evaluation metrics: the quality of spatial clustering is measured by the number of clusters generated by each algorithm, and also the total number of messages used for spatial clustering. To prove the approximation performance of HSC algorithm, we evaluate the energy consumption and average absolute error (*AAE*) of collected data when HSC is exploited by certain data collection scheme to conduct approximate data collection. In this paper, we substitute messages for energy consumption as data communication is the dominating energy consumer. Besides, *AAE* is calculated using Eq.(3), where r_i^t and c_i^t are the real value and the collected value through approximate data collection of sensor node s_i at the t-th epoch respectively. Moreover, N is the total number of sensor nodes and T is the total epoches of data collection.

$$AAE = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}|r_i^t - c_i^t|}{N \times T} \tag{3}$$

3) Parameters setting: N sensor nodes are deployed in a 100×100 sensing field, and Sink is located at $(120m, 50m)$. Communication radius R sets to be $30m$ for all sensor nodes. A DCT is pre-built with a simple method, i.e., by circulating a tree formation message originated by Sink and making use of a min-hop parent selection strategy. Note that other more

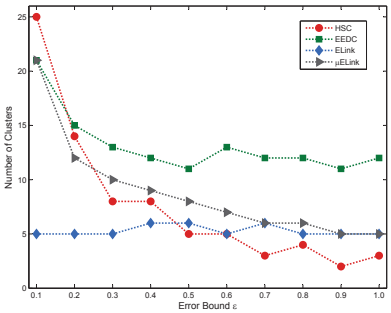

Fig. 2. Impact of ε on clustering quality

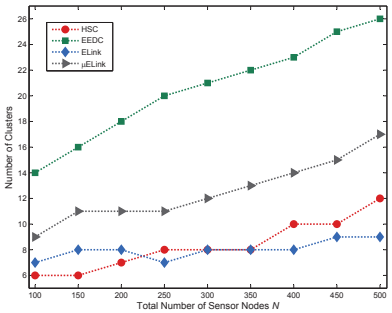

Fig. 3. Clustering quality with various N

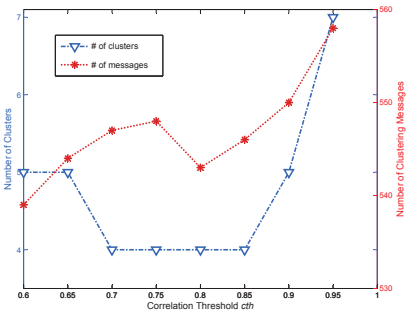

Fig. 4. Impact of cth on HSC clustering

efficient algorithms, e.g., [15], could be adopted. The size of the queue which contains the latest data for *AR* model learning sets to be $W = 60$, which is proved to be an appropriate value to produce a preferable *AR* model for environmental monitoring [8]. For the monitoring algorithm to maintain a dynamic local *AR* model, we adopt similar parameters setting like PAQ [8], i.e., $\theta = 0.03$, $\Lambda = 15$, and $\nu = 8$. Similarly, we adopt the optimum parameters setting for the compared algorithms. Explicitly, trend similarity threshold $t = 80\%$ and $gmax\_dst = 3\ hops$ for EEDC. Regarding to ELink and μELink, we set weight vector $\vec{w} = (0.5, 0.3, 0.2)$ for coefficients of *AR*(3) model, and fix the feature distance threshold $\delta = 0.5$. For our simulation study, we generate synthetic data set with similar method like DClocal [5]. 25 event sources are fixed in the sensing field in uniform distribution, readings of sensor node are comprehensive influencing results of all event sources, and the influence of an event source, say e_w, to node s_i is inversely proportional to the geographic distance between them. To simulate the values of event sources, we employ the publicly available Intel Lab dataset[1] which consists of 54 sensor nodes to measure various attributes, such as temperature, humidity, light and voltage. Owning to some data missing, We restore the temperature values of 51 nodes on March 9, 2004. At the beginning of each simulation, we randomly select 25 data series from the 51 nodes to map to the 25 event sources. Lastly, note that all results in this section are the average values of 10 simulations.

B. Quality of Spatial Clustering

In this sub-section, we study the impacts of system parameters, namely, user-defined error bound ε, correlation threshold cth and the total number of nodes N, on the clustering quality with comparisons to the three alternative algorithms. Fixing other parameters and setting $cth = 0.9$ for HSC, Fig.2 shows the total formed spatial clusters by each algorithm versus ε with $N = 100$. When ε increases from 0.1 to 1.0, the total cluster numbers of other three algorithms decrease except ELink. In most cases, HSC performs much better than the other algorithms. As ELink only takes the features computed on coefficients of *AR* model as the metric to measure similarity, the cluster number of ELink retains around 5. Moreover, we study the clustering quality in various network sizes by varying

[1]Intel lab dataset: http://db.csail.mit.edu/labdata/labdata.htm.

TABLE I
COMMUNICATION MESSAGES FOR SPATIAL CLUSTERING

	HSC	EEDC	ELink	μELink
$\varepsilon = 0.1, N = 100$	681	20515	2380	2544
$\varepsilon = 0.5, N = 100$	552	20643	2421	2470
$\varepsilon = 0.5, N = 500$	2649	95941	55291	55284

[a] for these simulation results, we set $cth = 0.9$ for HSC.

N with other parameters fixed as $\varepsilon = 0.5$ and $cth = 0.9$ for HSC, and the results are presented in Fig.3. It is easy to understand that with the increase of N, the whole network will be partitioned into more clusters. HSC still generates much fewer clusters than EEDC and μELink, and comparable quantity to ELink. As a matter of fact, merely considering the similarity on feature space but ignoring the magnitude similarity between nodes, ELink has the loosest similarity requirements among all the four algorithms. Particularly, we perform experiments to study the impact of cth on clustering quality of HSC with ε fixed as 0.5. Fig.4 shows that cth does affect the clustering quality. Explicitly, with more rigorous on correlation between nodes during similarity measure in HSC, more clusters will be generated, and more communication messages are needed accordingly. Obviously, when cth becomes greater, it will be more difficult for nodes to be similar with each other, thus more clusters are needed to cover all nodes.

Regarding to the communication messages for clustering, Table I presents several brief results. Undoubtedly, EEDC, due to transmitting all raw readings to Sink to perform centralized clustering, generates the most communication messages. Both ELink and μELink clustering are initiated by some well distributed sentinel nodes, thus they need vast of messages to ensure communicating correctly in an asynchronous network. Note that it would also be quite difficult to find such kind of sentinel nodes in practice. Relying on the DCT, our algorithm starts from Sink node and extends spatial clustering top down. It is illustrated as the results in Table I that HSC generates the fewest communication messages in various cases. Moreover, HSC performs spatial clustering based on the pre-built DCT rather than trying to build routing tree based on the formed clusters. Therefore, HSC could be used together with some optimal DCT constructing algorithms to make data collection more effective and efficient.

TABLE II
AVERAGE ABSOLUTE ERROR

	HSC	EEDC	ELink	μELink
AAE	0.082	0.081	0.132	0.103

C. Efficiency of Spatial Clustering

To prove the efficiency of spatial clustering, we have performed simulative approximate data collection by combining with the four spatial clustering algorithms. In following simulations, we set parameters $\varepsilon = 0.5$, $cth = 0.9$ for HSC, and perform 1000 epoches data collection for all algorithms. For EEDC, we adopt the same randomized intracluster scheduling and data restoration method [3] to perform approximate data collection. For other three algorithms, we adopt the centralized model of PAQ [8] to perform approximate data collection, i.e., Sink predicts readings for each sensor node with values of its corresponding clusterhead, which sends periodic readings to Sink. As is shown in Fig.5, HSC and ELink consumes much fewer communication messages for approximate data collection than other two algorithms. Specifically, EEDC needs the most messages for data collection term as centralized EEDC needs all nodes in the same cluster to transmit data to Sink when it detects dissimilarity in a cluster. μELink consumes the most messages for clustering term, which includes messages for spatial clustering and cluster maintaining. Because of the introduction of baseline value and the distinctive distribution of clusters, μELink needs more messages to track the similarity between nodes. Furthermore, Table II presents the collected data error for each algorithm. EEDC has the smallest AAE due to its abundant data collection messages, and HSC takes the second place with a comparable result. Taking the baseline value μ into account, μELink surpasses ELink on this metric at the cost of huge cluster maintaining messages. In summary, HSC has got much better comprehensive performance on the efficiency of data approximation than other three algorithms.

VI. CONCLUSION

To perform efficient approximate data collection, we consider spatial clustering for WSNs. Taking the coefficients of *AR* model as an important clustering parameter, we design a novel similarity measure method which gives coequal consideration to both magnitude similarity and trend similarity. With this measure method, our algorithm forms spatial clusters based on DCT in a distributed and hierarchical manner to get rid of requirements on strict time synchronization and global topology information. Simulation results with typical data set show the superior clustering quality of HSC when compared with three alternative algorithms. Furthermore, simulations based on a simple approximate data collection scheme illustrate the efficiency and accuracy of our algorithm in data collection.

ACKNOWLEDGMENT

This work is partially supported by Scientific and Technical Innovation Team Project of Zhejiang Province for Digital Culture and Multimedia Technology (No.2010R50040).

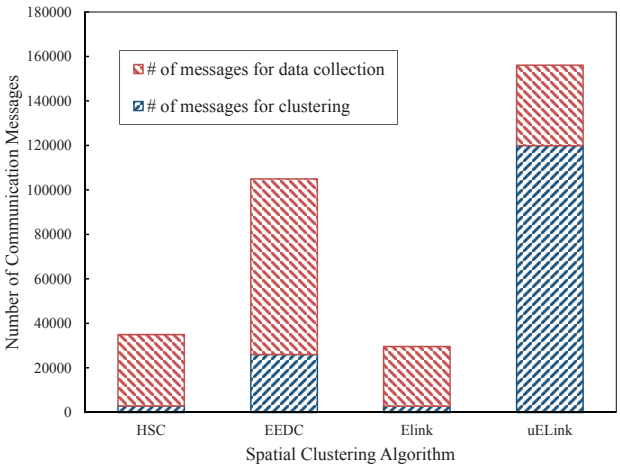

Fig. 5. Communication messages for approximate data collection

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] C. Wang, H. Ma, Y. He, and S. Xiong, "Adaptive approximate data collection for wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 6, pp. 1004–1016, 2012.

[3] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 7, pp. 1010–1023, 2007.

[4] C. Hung, W. Peng, and W. Lee, "Exploiting spatial and data correlations for approximate data collection in wireless sensor networks," in *Proceedings of the Second International Workshop on Knowledge Discovery from Sensor Data (Sensor-KDD)*, 2008.

[5] ——, "Energy-aware set-covering approaches for approximate data collection in wireless sensor networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 11, pp. 1993 –2007, nov. 2012.

[6] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2006.

[7] A. Meka and A. Singh, "Distributed spatial clustering in sensor networks," in *International Conference on Extending Database Technology (EDBT)*, 2006.

[8] D. Tulone and S. Madden, "PAQ: time series forecasting for approximate query answering in sensor networks," in *the European Conference on Wireless Sensor Networks (EWSN)*, 2006.

[9] A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[10] C. Huang, J. Huang, J. Yan, and L. Yeh, "An in-network approximate data gathering algorithm exploiting spatial correlation in wireless sensor networks," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC)*, 2012.

[11] S. Bahrami, H. Yousefi, and A. Movaghar, "Daca: data-aware clustering and aggregation in query-driven wireless sensor networks," in *the 21st IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2012.

[12] Z. Liu, W. Xing, B. Zeng, Y. Wang, and D. Lu, "Distributed spatial correlation-based clustering for approximate data collection in WSNs," in *the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2013.

[13] J. Peter and A. Richard, *Introduction to time series and forecasting, 2nd edition*. Springer, 2002.

[14] Z. Liu, W. Xing, Y. Wang, and D. Lu, "An energy-efficient data collection scheme for wireless sensor networks," in *the 15th International Conference on Advanced Communications Technology (ICACT)*, 2013.

[15] J. Liang, J. Wang, J. Cao, J. Chen, and M. Lu, "An efficient algorithm for constructing maximum lifetime tree for data gathering without aggregation in wireless sensor networks," in *IEEE INFOCOM*, 2010.