International Journal of Smart Sensor and Adhoc Network

Volume 2 | Issue 3 Article 4

April 2012

AMORPHOUS EVENT PREVENTION IN WIRELESS SENSOR NETWORK

V. SURENDHIRAN
Department of Computer Science, and Engineering, Angel college of Engineering and Technology,Tirupur, surendhiran.svs@gmail.com

S. SABARISH
Department of Computer Science, and Engineering, Angel college of Engineering and Technology,Tirupur, samsha.sam800@gmail.com

P. PREMADEVI
Department of Computer Science, and Engineering, Angel college of Engineering and Technology,Tirupur, prema.acetcse@gmail.com

AMORPHOUS EVENT PREVENTION IN WIRELESS SENSOR NETWORK

V.SURENDHIRAN[1], S.SABARISH[2] & P.PREMADEVI[3]

[1,2&3]Department of Computer Science, and Engineering, Angel college of Engineering and Technology,Tirupur
Email: surendhiran.svs@gmail.com, samsha.sam800@gmail.com, prema.acetcse@gmail.com

Abstract -In many applications wireless sensor can be used to detect the events in those applications. With the advances in sensing, communication, and computation, there is an increasing need to track mobile events such as air pollutant diffusion, toxic gas leakage, or wildfire spreading using mobile sensors such as robots. Lots of existing work use control theory to plan the path of mobile sensors by assuming that the event evolution is known in advance. This assumption has severely limited the applicability of existing approaches. In this paper we aim to design a detecting, tracking and preventing approach that is capable of identifying multiple events with dynamic event signatures and providing event evolution history that may include event merge, split, create and destroy. We also focused on the power consumption.

1. INTRODUCTION

Mobile sensor networks are very powerful when being used to detect and track mobile events such as air pollutant diffusion, toxic gas leakage, or wildfire spreading. However, existing work like assumes that event evolution is known in advance so that events can be modeled formally and robots can be controlled according to track the events. This assumption has severely limited the applicability of existing approaches, especially in a general scenario containing multiple dynamic events with different evolving patterns. To save energy resource and thus extend the network lifetime, it is desirable that only the nodes that surround the mobile target are responsible for observing the target. For example, when the target passes through the t1 point as shown in Fig. 1, all nodes do not need to join in the task for tracking a mobile target. Instead, it is more energy efficient for only the nodes S1 around the mobile object to join in collecting information of the target and performing collaborative work among them. Other nodes located far from the target do not need to waste their powers to monitor the target. If we can predict the next location of the mobile object in advance, we can organize the group membership dynamically which should join in tracking mission. As shown in Fig.1 for example, the number of participating nodes may be minimized, which allows us to further extend the whole network lifetime if we predict future location of the mobile target accurately As the mobile object moves, the tracking nodes may migrate to the moving direction of the target to keep on monitoring as shown in Fig. 1, where a thick line indicates the moving path of the mobile target and the blacked circles inside the dotted circle are tracking nodes at time t1 . Thus, sensor nodes need to control their states by themselves based on prediction of target's movement. We assume a sensor network where N sensors with the same communication and sensing range are distributed randomly in the environment that is being monitored. We also assume that each node knows its own location by using GPS or other location awareness techniques.

And we utilize triangulation for localization of a mobile target. Consequently, at least 3 sensors join the target detection and tracking with surveillance. Also each node keeps information about its neighbors such as location through the periodically message change. And each individual sensor node is equipped with appropriate sensory devices to be able to recognize the target as well as to estimate its distance based on the sensed data. Further, we assume that we predict the location of the mobile targets every one second (or minute), and each sensor records the movement pattern of the mobile object. Basically, we use a moving average estimator to predict the future location of the mobile target based on the measurement of direction and the velocity of the mobile target. WIRELESS sensor networks have been considered very useful for event detection and tracking in various applications such as oil spill detection or ground water contaminant monitoring. The challenge here is to devise a method for the sensors to recognize and follow these events as they travel through the network. This identification and tracking capability forms a critical foundation for various higher level processing tasks such as predicting an event's evolution or conducting queries on the affected area. For instance, for some applications like monitoring the dispersion of fluids, classic numerical fluid transport models for fluid prediction are extremely computationally intensive and require hours to run to completion. In order to monitor events in real time, the model should be decomposed and computation should be distributed among the sensor nodes to exploit computational parallelism. By identifying and tracking each event in a distributed manner, one node for each identified event can be designated as an interface point for running the model. Typical examples include establishing survivable military surveillance systems, environmental and industrial monitoring, personnel and wildlife monitoring systems requiring tracking schemes, capable of deducing kinematic characteristics such as position, velocity, and acceleration of single or multiple targets of interest. The basic idea of our tracking approach is as follows. An entity that intends to track a target is called a tracker. A tracker is assumed to be a single generic source such as a mobile user or a respective authority. A target can be any mobile entity such as enemy vehicle or an intruder. Each sensor in the network has the capability of sensing, communicating, and computing. Consider again the chemical spill as it diffuses below ground. If the fluid is pouring out from more than one site, the separate plumes may meet and mix together. In so doing, they lose their individual shapes in a single large cloud. Conversely, changes in the medium through which it permeates may cause the fluid to follow a few preferred paths and break up into separate, smaller concentrations. In practice, keeping track of the dynamic expanding, shrinking, dividing, and merging of contaminant is essential to making treatment decisions. In edge detection-based region event tracking, the challenge is to devise a method for nodes to be identified as "edge nodes" that are near the boundary of a region and from that, calculate an approximate boundary for the region in question. Three methods guided by statistics, image processing techniques, and classifier technology are developed and compared in. A novel method for edge detection of region events makes use of the duel-space principle. The algorithm is fundamentally centralized. Identify several critical points in a given event region and periodically check the criticalness of these points, but the scheme can only work for an event whose shape remains convex. Therefore, we have the realization of the aforementioned intuition that if a high concentration of an event's readings is moving far off the event center, then that concentration should be recognized as an autonomous event. Merges are symmetric in logic. The node momentum is the decision variable that controls whether two events should remain logically distinct or instead be folded into one entity. The possible outcomes of this decision control the event splitting and merging powers unique to our proposed solution. In a naive approach, a split occurs when a group of nodes have momenta with magnitudes above a certain value. This, however, is insufficient. In real situations, nodes will have high magnitude momenta naturally if an event is simply very large. A simple momentum threshold limits the size of events that can be detected. What needs to be determined instead is if a node's momentum is large relative to the event's overall size. First, we present some concepts that illuminate the distributed nature of our protocol. If an event is entirely contained within one cluster, then that clusters head can run DRAGON locally in a centralized manner. A foremost need is to allow cluster heads to take counsel with each other for cases where an event spans multiple clusters. Also, there is a need for global orchestration when deciding which existing events may be merged. To this end, we discuss the concept of the backbone tree to facilitate cooperation and to control DRAGONs execution throughout the network.

2. RELATED WORK

DRAGON proposes general purpose event detection and tracking algorithm that is capable of identifying dynamic events even in the presence of event splits and merges. However, DRAGON works for stationary wireless sensor networks, which are not practical for some applications such as contaminant cloud monitoring where sensors become mobile due to winds. Also, a large number of sensor nodes will be needed when the detection area grows larger and larger. To address these issues, this work investigates the use of mobile sensor networks for dynamic event detection and tracking. DRAGON employs two physics metaphors: event center of mass, to give an approximate location to the event; and node momentum, to guide the detection of event merges and splits. Both detailed theoretical analysis and extensive performance studies of DRAGON's properties demonstrate that DRAGON's execution is distributed among the sensor nodes, has low latency, is energy efficient, is able to run on a wide array of physical deployments, and has performance which scales well with event size, speed, and count. They will use sensors and robots interchangeably in the same way which we are going to use.

Disadvantages:

1) DRAGON does cost more energy than R-DCTC due to the nature of the expanded problem.
2) DRAGON cannot easily compete directly with R-DCTC in terms of time complexity
3) There is no prevention method in the previous system.

3. FOUNDATION CONCEPT

This paper presents MEMS—a novel pipelined approach for dynamic event detection and tracking. With rapid advances in sensor fabrications, recent sensors are designed to be power aware, changing their condition (e.g., shut down sensing processor or radio) when they do not need to run the components to perform a given task in a sensor field. Most sensors can operate under the three different conditions: Active, Idle and Sleep. It is important to completely shut down the radio rather than put it in the idle mode when it needs not sensing. Power management of sensor components is very important because energy consumption depends on their duties. In the detection phase, each detection robot follows a certain path in their detection units to check any new event regions (i.e., consecutive sensing cells that are detected with events). At the end, all the information resides in one detection robot and that particular robot will be responsible for sharing the event region with other robots in the future. In the tracking phase, detection robots assign each event region to several tracking robots, where the number of tracking robots is determined by the event region size and the robot speed. We also implement prevention method.

4. APPROACHES

4.1 Detection robots in a distributed way Each detection robot follows a certain path in their detection units to check any new event regions (i.e., consecutive sensing cells that are detected with events). For instance, the robot first moves rightwards all the way to the boundary of the detection unit, then downwards to the adjacent sensing cell, and then leftwards all the way to the boundary of the detection unit, and then downwards to the sensing cell below. After this step, the detection robots have clear ideas which sensing cells are within the event regions. If an event region is only inside one detection unit, then the corresponding detection robot has the complete information of the event region in terms of the space the event region occupies. Otherwise, if the event region spans several detection units, the corresponding detection robots in those units need to consolidate their information about the event region and designate one detection robot to hold the complete information of the event region. MEMS accomplish this by gathering the information from all relevant detection units in a clock-wise fashion. At the end, all the information resides in one detection robot and that particular robot will be responsible for sharing the event region with other robots in the future. During the simulation, the events move individually with varying direction and speed no larger than the maximum speed in the detection area until merges or splits happen. Once a merge happens, the events merged into one event will have the same movement pattern. Once a split happens, the events will have individual movement patterns. Also, there are certain chances of event creation and event destroy in each round. Energy detection uses minimal a priori information about the target. The detector essentially computes a running average of the signal power over a window of pre-specified length. The output of the detector is sampled at a pre-specified rate. The window duration and sampling rate are determined by target characteristics, such as the signature bandwidth and the expected signature duration in the particular sensing modality. An event is detected when the detector output exceeds a threshold. Due to the inherent signal averaging, the noise component in the output of the detector may be modeled as a Gaussian random variable whose mean and variance can be determined from the statistics of the background noise. 4.2 The minimal number of tracking robots as needed The detection of robots assign each event region to several tracking robots, where the number of tracking robots is determined by the event region size and the robot speed. Further, detection robots plan the tracking path according to the consecutive event regions assigned to the tracking robots. Tracking robots sense the events along their tracking paths, and find event entry and exit boundary point and send the information to detection robots.

A O(nlog(n)) plane sweep algorithm is applied to the boundary point pairs to separate the individual events in each event region. The problem of tracking targets with sensor networks has received attention from various angles. We consider the case where a set of k targets need to be tracked with 3 sensors per target from the resource requirement viewpoint. They show that the probability that all targets can be assigned 3 unique sensors shows phase transition properties as the level of communication between the sensors increases. In an information driven sensor collaboration mechanism is proposed. In this mechanism, measures of information utility are utilized to decide future sensing actions. Collaborative signal processing aspects for target classification in sensor networks is addressed. Tracking based on relations in the targets. Techniques for locating targets using a variety of mechanisms have been proposed. However, current literature does not address the issue of a scalable architecture for coordinating a sensor network for the purpose of target tracking. Nor is there any existing work which deals with the feasibility, minimization of computation and communication overheads and understanding the tradeoffs in such systems. In this paper we address these issues. To be effective, the tracking system should be accurate and the likelihood of missing a target should be low. Additionally, the dynamic range of the system should be high while keeping the response latency, sensitivity to external noise and false alarms low. The overall architecture should also be robust against node failures. Tracking multiple targets via a wireless sensor network is a very challenging, multi-faceted problem and several research groups have tackled various aspects of it. We consider the signal processing aspects of this problem under the constraints imposed by limited capabilities of the nodes as well as those associated with networking and routing. Consequently, in the present form, all our algorithms are based on processing a single sensing modality, such as seismic or acoustic. Furthermore, current detection and classification algorithms are based on single-node processing, whereas localization and tracking algorithms require collaboration between nodes. Our main emphasis in this paper is on target classification that is arguably the most challenging signal processing task in the context of sensor networks. We provide some numerical results based on real data that lend useful insights into the problem and help identify key issues and challenges. Finally, based on our findings we identify some promising directions for future research.

4.3 Identifies multiple events with dynamic event signatures Identifying multiple events with dynamic event signatures and providing event evolution history that may include event merge, split, create and destroy. MEMS provides event signature with a label consisting of round number, detection robot ID, and the group ID of the corresponding tracking robots. If multiple targets are sufficiently separated in space or time, that is they occupy distinct space-time cells. It may be used: a different track is initiated and maintained for each target. Sufficient separation in time means that the energy detector output of a particular sensor exhibits distinguishable peaks corresponding to the CPAs of the two targets. Similarly, sufficient separation in space means that at a given instant the spatial target signatures exhibit distinguishable peaks corresponding to nodes that are closest to the targets at that instant. The assumption of sufficient separation in space and/or time may be too restrictive in general. In such cases, classification algorithms are needed that operate on spatio-temporal target signatures to classify them. This necessarily requires a priori statistical knowledge of typical signatures for different target classes. In this paper, we focus on single-node (no collaboration between nodes) classification based on temporal target signatures: a time series segment is generated for each detected event at a node and processed for classification. Some form of temporal processing, such as a fast Fourier transform (FFT), is performed and the transformed vector is fed to a bank of classifiers corresponding to different target classes. The outputs of the classifiers that detect the target, active classifiers, are reported to the manager nodes as opposed to the energy detector outputs. The object corresponds to tracking the location of the spatial peak over time. To enable such tracking in a sensor network, the entire space-time region must be divided into space-time cells to facilitate local processing. The size of a space-time cell depends on the velocity of the moving target and the decay exponent of the sensing modality. It should approximately correspond to a region over which the space-time signature field remains nearly constant. In principle, the size of space-time cells may be dynamically adjusted as new space-time regions are created based on predicted locations of targets.

4.4 Event evolution

Event evolution contains a series of records of the dynamic event signatures and the event merge/split/create/destroy actions in each round. Event evolution tree is constructed to evolution. Based on the event evolution tree, we can conduct event queries to show the events evolution history.

This allows us to take the data distribution models of two different sensors in the network and construct a single model that describes the behavior of the data of both sensors. Our kernel estimators can be easily combined, and thus are well suited for our framework. There are two quantities that we have to combine, the sample set, R, and the bandwidth of the kernel function, B. We can combine the sample sets just by taking their union. We may then reduce the size of the resulting set by re-sampling, if necessary. In order to combine the bandwidths of two kernel functions, we only need to combine the two standard deviations upon which the bandwidths depend. This is accomplished using the same techniques as the ones for computing the standard deviation in a sliding window of streaming data. This method demands high amounts of energy since it requires transmission of the kernel samples and the bandwidth from each sensor to the sink. Moreover, it incurs high latency in transmission due to the large number of packets sent across the network. We propose a distributed technique, where we detect homogeneous regions at each cell in the grid, and then communicate only the summary information of each cell to the leader in the next higher level in the network.

5. AMORPHOUS EVENT PREVENTION

Once a split happens, the events will have individual movement patterns. Also, there are certain chances of event creation and event destroy in each round. Amorphous events happened at the time of splitting hence we prevent that event. Search engine speeding: Almost all webmasters value any and all attention they receive from search engines. Some businesses run solely on search engine rankings and the visitors they get from those sources. So, these techniques should in no way affect the ability of these “automated spiders” to spider the website effectively. Since the scanners and these search engine spiders would be automated, differentiating them would be difficult. Sure, the user agents would give away their identities, but then again forging user agents isn’t a very difficult task, thus rendering that method useless. However a very big difference in the way search engines and scanners spider is their intent. Search engines, aim to please webmasters and thus follow the instructions in the “robots.txt” file, as opposed to scanners which tend to use the robots.txt as a place to find hidden and sensitive links. This would be a perfect way to create a honey pot for these scanners while allowing the search engines to spider harmlessly.

6. CONCLUSIONS

In this work, we have presented MEMS, general purpose event detection and tracking algorithm that is able to operate in the presence of event splits and merges. MEMS has been shown to be highly accurate across a wide range of scenarios. It consistently finds the right number of events and outlines the right event shapes regardless of deployment type, and regardless of event size, speed, or count. MEMS's energy efficiency scales well with problem size and complexity. The energy cost's order of growth is always shown to be linear or better with respect to the number of events. DRAGON's execution time is projected to be well within the constraints necessaryto keep up with virtually any kind of event. Overall, DRAGON is promising for applications using wireless sensor networks for phenomena monitoring.

REFERENCES

[1] Nicholas Hubbell and Qi Han," DRAGON:Detection and Tracking of Dynamic Amorphous Events in Wireless Sensor Networks ", IEEE Transactions on Parallel and Distributed Systems, VOL. 23. No.7, JULY 2012.

[2] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a Moving Object with a Binary Sensor Network," Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys), pp. 150-161, 2003.

[3] R.R. Brooks, P. Ramanathan, and A.M. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks," Proc. IEEE, vol. 91, no. 8, pp. 1163-1171, Aug. 2003.

[4] C. Buragohain, S. Gandhi, J. Hershberger, and S. Suri, "Contour Approximation in Sensor Networks," Proc. Int'l Conf. Distributed Computing in Sensor Systems (DCOSS), pp. 356-371, 2006.

[5] K.K. Chintalapudi and R. Govindan, "Localized Edge Detection in Sensor Fields," Proc. Int'l Workshop Sensor Network Protocols and Applications (SNPA), pp. 59-70, 2003.

[6] S. Gandhi, J. Hershberger, and S. Suri, "Approximate Isocontrours and Spacial Summaries for Sensor Networks," Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN), pp. 400- 409, 2007.

[7] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J.A. Stankovic, T. Abdelzaher, and B.H. Krogh, "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments," Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys), pp. 205-217, 2005.

[8] N. Hubbell, "Dragon: Detection and Tracking of Amorphous Events with Dynamic Signatures in Wireless Sensor Networks," master's thesis, Colorado School of Mines, http://www.mines. edu/~qhan/research/techrep/nhubbell-thesis.pdf, 2009.

[9] H. La and W. Sheng, "Flocking Control of a Mobile Sensor Network to Track and Observe a Moving Target," Proc. IEEE Int'l Conf. Robotics and Automation (ICRA), 2009.

[10] L. Lazos, R. Poovendran, and J.A. Ritcey, "Probabilistic Detection of Mobile Targets in Heterogeneous Sensor Networks," Proc. Int'l Conf. Information Processing in Sensor Networks (IPSN), pp. 519-528, 2007.

[11] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and Scalable Simulation of Entire Tinyos Applications," Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys), pp. 126-137, 2003.

[12] M. Li and Y. Liu, "Underground Coal Mine Monitoring with Wireless Sensor Networks," ACM Trans. Sensor Networks, vol. 5, no. 2, 2009.