MEMS: Detection and Tracking of Mobile Eevents Using Mobile Sensors

Na Yu and Qi Han
Department of Mathematical and Computer Sciences
Colorado School of Mines, Golden, CO 80401
{nyu, qhan}@mines.edu

***Abstract*—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 work, we aim to design a detection and tracking 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. Simulation results show that our approach can identify events with low event count difference, high event membership similarity, and accurate event evolution decisions, while using a reasonable number of tracking robots.**

***Index Terms*—event detection; event tracking; event evolution; robotic sensor networks**

I. 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 [1][2][3] 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.

For many applications, it is essential for an event tracking approach to be able to recognize, label, and follow dynamic events. This is because event evolution history can be used to analyze event patterns and predict future behaviors, resulting in more informed decision making or remediation plans. Previous work—DRAGON [4] proposes a 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. We will use sensors and robots interchangeably in the following.

This paper presents MEMS—a novel pipelined approach for dynamic event detection and tracking. The noteworthy features of MEMS are that it (1) uses detection robots in a distributed way; (2) uses the minimal number of tracking robots as needed; (3) accurately identifies multiple events with dynamic event signatures; (4) provides accurate event evolution history including event merge, split, create and destroy.

II. Design of MEMS

MEMS works in two phases: detection and tracking. To speed up the process, MEMS adopts a pipelined approach where detection and tracking execute in parallel (Fig. 1). MEMS periodically initiates a new round of detection, whose results will be used for tracking in the next round.

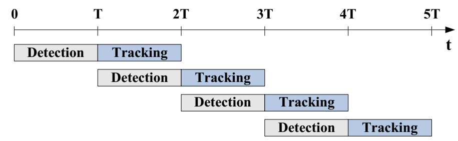


Fig. 1. Event detection and tracking pipeline

During initialization, the detection area is divided into detection units, with one detection robot in each unit. The size of a detection unit is determined in a way that the robots can directly communicate with robots in adjacent detection units. Each detection unit is further divided into a number of sensing cells. The size of a sensing cell is determined by the sensing range of the sensors used to track the events. The time period for one round is calculated based on the maximum event size, event speed, and robot speed.

After initializing the network, MEMS works as follows (Fig. 2). 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). 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

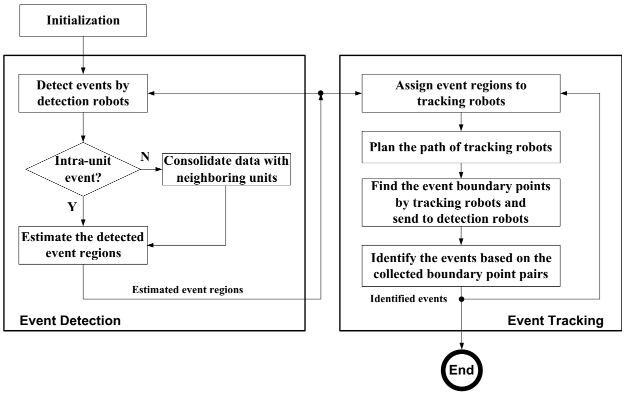


Fig. 2. Detailed event detection and tracking in one round

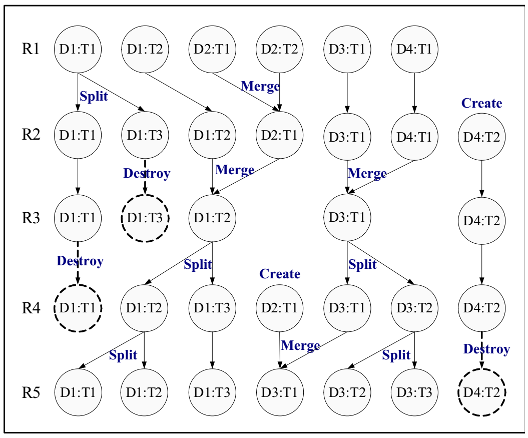


Fig. 3. Sample event evolution with dynamic event signatures

designate one detection robot to hold the complete information of the event region. MEMS accomplishes 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.

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. 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 points 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. Further, event properties such as event center of mass and event size are also compared with those in the last round to identify possible merges or splits.

A unique feature of MEMS is the concept of **event signatures**, the distinct identities of events. MEMS provides event signature with a label consisting of round number, detection robot ID, and the group ID of the corresponding tracking robots. Event evolution contains a series of records of the dynamic event signatures and the event merge/split/create/destroy actions in each round. An example event evolution tree is shown in Fig. 3, where R# means the round number, D# means the detection robot ID, and T# means the group ID of tracking robots. Based on the event evolution tree, we can conduct event queries to show the events evolution history. For example, query the event which has the signature $R3D1T2$, then the event history is shown as $R3D1T2 \leftarrow R2D1T2 + R2D2T1 \leftarrow R1D1T2 + R1D2T1 + R1D2T2$. We can further map the detection robot ID and tracking ID to geographical locations, which in conjunction with the contour maps provided by tracking robots will provide a clear idea of how the event has changed over time.

III. Evaluation of MEMS

We implement a middleware layer simulation environment focusing on the event detection and tracking rather than the underlying networks to evaluate MEMS.

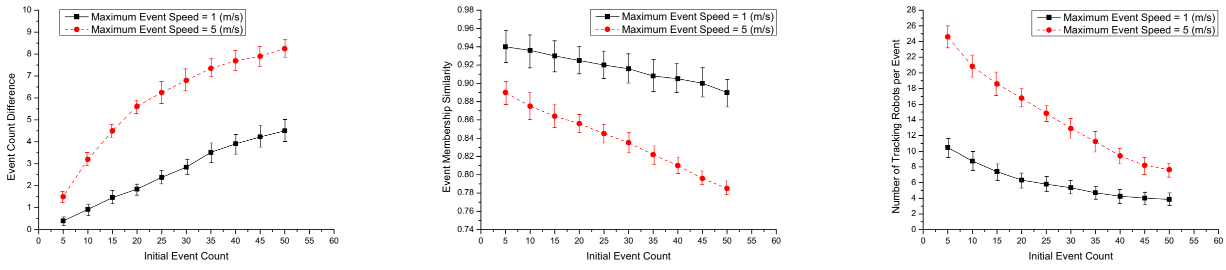
The performance metrics we consider are:

- Event Count Difference: the event count difference between the tracking results and the ground truth is used to show whether MEMS successfully detects all of the distinct events in the detection area.
- Event Membership Similarity: this metric is similar to the Jaccard Coefficient in data mining and is used to show whether each individual event is delineated correctly.
- Number of Tracking Robots per Event: the average number of tracking robots used to track an event during the event evolution is used to show whether MEMS uses a reasonable low number of tracking robots.
- Action Count: the number of rounds in 100 rounds that MEMS identifies an event merge/split/create/destroy is compared with the actual event actions to demonstrate the detection and tracking accuracy at a high level.

The parameters to evaluate the performance are:

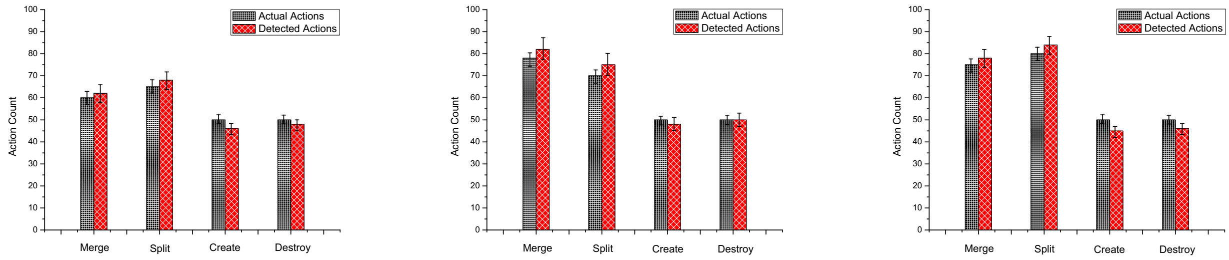
- Detection Area Size: the size of the entire detection area, which is represented in units*units.
- Maximum Event Size: the maximum size of an event that can have during evolution, which is represented in units.
- Maximum Event Speed: the maximum speed (m/s) of an event that can have during evolution.
- Initial Event Count: the existing number of events when the first round of event detection starts.

We model events using the parameters above, where the number of events equals the initial event count and are generated at the beginning of the simulation, each with a random event size no larger than the maximum event size and a random event speed no larger than the maximum event speed. 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.



(a) (b) (c)

Fig. 4. Impact of Initial Event Count



(a) InitialEventCount=20, MaximumEventSpeed=1 (b) InitialEventCount=20, MaximumEventSpeed=5 (c) InitialEventCount=40, MaximumEventSpeed=1

Fig. 5. Event Action Counts

Preliminary Results: We present a preliminary set of simulation results to show the impact of the initial event count with different maximum event speeds. The parameters are shown in TABLE I, and the corresponding results are shown in Fig. 4 and Fig. 5.

TABLE I
PRELIMINARY SET OF PARAMETERS

Parameter	Value Range
Detection Area Size	10*10 units
Maximum Event Size	10 units
Maximum Event Speed	1m/s, 5m/s
Maximum Robot Speed	10m/s
Initial Event Count	5 to 50

Fig. 4(a) shows that the event count difference increases when the initial event count increases. This is because for the same detection area size and maximum event size, as the initial event count increases, the event count is more likely to be larger during the event evolving. Fig. 4(b) shows that the event membership similarity decreases when the initial event count increases due to more independent event movements. Fig. 4(c) shows that the number of tracking robots per event decreases when the initial event count increases, and it depends much on the maximum event speed. In conclusion, MEMS provides accurate event detection and tracking when the number of events in the area is small; and its performance degrades as the number of events or the event speed increase. Fig. 5 shows that at a high level MEMS detects the similar number of event changes as the ground truth. With more events in the area or faster events, there are more event merges and splits.

IV. CONCLUSION

In this paper, we have presented MEMS, a novel pipelined approach for detecting and tracking events with dynamic event signatures. MEMS has been shown to be capable of identifying multiple events with low event count difference and high event membership similarity, while using a reasonable number of tracking robots. Further, it has also been verified to provide accurate event evolution history including event merge, split, create and destroy. Overall, MEMS is promising for phenomena monitoring applications using robotic sensor networks.

ACKNOWLEDGEMENT

This project is supported in part by NSF grant CNS-0915574.

REFERENCES

[1] A. Sinha, A. Tsourdos, and B. White, "Monitoring the dispersion of a contaminant cloud in an urban region by a swarm of uav sensors," in *Proceedings of The IFAC Workshop on Networked Robotics*, 2009.
[2] S. Subchan, B. A. White, A. Tsourdos, M. Shanmugavel, and R. Zbikowski, "Dubins path planning of multiple uavs for tracking contaminant cloud," in *Proceedings of the 17th World Congress*, 2008.
[3] B. A. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Zbikowski, "Contaminant cloud boundary monitoring using network of uav sensors," *IEEE Sensors Journal*, vol. 8, no. 10, 2008.
[4] N. Hubbell and Q. Han, "Detection and tracking of dynamic amorphous events in wireless sensor networks," in *Proceedings of SECON*, 2011.