A Decentralized Approach for Detecting Dynamically Changing Diffuse Event Sources in Noisy WSN Environments [1]

Jose Luis Fernandez-Marquez, Josep Lluis Arcos
Artificial Intelligence Research Institute, IIIA
Spanish National Research Institute, CSIC
Campus UAB, Bellaterra, E-08193 (Spain)
{fernandez,arcos}@iiia.csic.es

Giovanna Di Marzo Serugendo
Birkbeck College
University of London
Malet Street WC1E 7HX London, UK
dimarzo@dcs.bbk.ac.uk

***Abstract*—Localizing dynamically changing diffuse event sources in real environments is still an open problem in Wireless Sensor Networks (WSN). The dynamism of the environment, the energy limitations of the sensors, and the noise associated to the sensor's measurements is a challenge that a realistic solution has to deal with. In this paper we propose a decentralized approach to detect diffuse event sources in dynamic and noisy environments, using a Wireless Sensor Network infrastructure and following a gradient-based strategy.**

I. Introduction

The localization of diffuse event sources and plumes is a problem that appears in a wide range of real applications such as toxic gas, underwater leaks, acoustic and heat sources detection. Diffuse events are huge phenomena that can spread in a 2D or 3D space without a regular shape. A diffuse event consists of one *source*, the focus of the event, and its *plume*, the area the diffuse event covers. Plume sizes and shapes are constantly changing due to the environment dynamism that acts over them (e.g. wind, obstacles).

Approaches exploiting WSN, have essentially concentrated on detecting plumes using centralized algorithms, on detecting a single source (global optimum) in static and noise-free environments [1], or detecting multiple sources with sensors well distributed in the environment and following a centralized strategy [2]. Ruair et al. [3] demonstrated that existing algorithms for target tracking do not scale well when they are applied to the localization of diffuse event sources. These algorithms require that each sensor reports the data to the sink when it reads a sensor value higher than a threshold. Since diffuse events can cover large areas, a large number of sensors would try to report the data to the sink, producing a network overload. In order to avoid this problem, this paper focuses on detecting the diffuse event sources as a first step before reporting the information to the sink. Thus, only a few number of sensors, those placed at the sources, will report the information to the sink. To the best of our knowledge, the problem of detecting dynamically changing diffuse event sources in noisy WSN environments has not been addressed before.

[1] Work funded by TIN2009-14702-C02-01, CONSOLIDER CSD2007-0022, and CSIC-PIF08-15-2 projects.

II. Our Approach

The aim of our approach is to localize the diffuse event sources as soon as possible, minimizing sensor measurements and communication. We propose a multi-agent approach based on mobile agent technology where sensors are responsible for creating agents and to provide an infrastructure to host agents allowing the agents to access their data and communication devices. Sensors are most of the time in the sleep state, that is, with the wireless communication turned off and using low energy. Every T_w ticks, a sensor creates an agent with probability P_a. Mobile Agents are responsible to actively track diffuse event sources and to monitor them once they reach the source. Mobile Agents use the WSN to move over the space, to obtain sensor data, and to communicate with other sensors or agents.

When agents are created, they try to reach the closer diffuse event source by following the shortest path according to a gradient-based strategy. Specifically, each agent uses the sensor data of the neighboring sensors to guide its movements and to find the source. Following Algorithm 1, when an agent is created, it first checks if there is another agent placed in one of the adjacent sensors. If that is the case, the most recent agent finishes its execution. Otherwise, it reads its host sensor data and it checks if a given event plume is detected. If nothing is detected (the measured value is too low), it finishes its execution. When an event is detected, the execution continues by choosing n_s adjacent sensors and sending a sensor data request to the selected n_s sensors. When all the answers are received, the sensor providing the highest sensor data read is selected. If the data of the best neighbor sensor is higher than the data the agent has measured on its host sensor, the agent migrates to the selected sensor. After migrating, if another agent is already hosted at that sensor, the migrating agent finishes its execution. Otherwise, the main loop starts again (reading the sensor data of the host sensor).

When an agent reaches the source of a diffuse event (i.e. when it does not move between consecutive reads), it starts sending data to the sink, and then continuously monitors the

978-0-7695-4232-4/10 $26.00 © 2010 IEEE
DOI 10.1109/SASO.2010.30

Algorithm 1 The Agent Algorithm

```
if (agentsInNeighborhood()) then exit() end
while (true) do
    sensorData = readSensor()
    if (sensorData <= 0) then exit() end
    neighbors = selectAdjNodes (n_s)
    requestReads (neighbors)
    bestSensor = selectBestSensor (neighbors)
    if (bestSensor.data > sensorData) then
        moveToSensor(bestSensor)
        if (existAgentInSensor() ) then
            exit()
        end
    end
end
```

event until an environment change occurs. An event source may disappear or change its location. When it disappears, the data obtained from the sensor becomes zero and the agent finishes its execution. When an event source changes its position (i.e. the event moves slightly), the data obtained from the neighbor sensors will guide the agent to the new source location.

III. Experiments

We conducted several experiments to validate our approach. The parameter settings were: a simulation time $T_s = 2 \times 10^5$ ticks, a frequency of environment changes $t_c = 200$ ticks, where an environment change consists in a new set of diffuse event sources from 1 to 3, the number of sensors $S_N = 10^3$, the frequency of the agent creation $T_w = 20$ ticks, the probability of creating an agent $P_a = 0.5\%$, and the number of nearby sensors receiving a data request from an agent $n_s = 3$. That is, a simulation holds 10^3 environment changes. The results reported are the averages of these 10^3 changes.

The number of data sensor reads and the number of msgs sent were used as an estimation of the cost to reach the convergence, i.e. when *all* diffuse event sources of a given scenario have been detected. We consider a failure of the system if the system cannot reach the convergence before a new change in the environment (i.e. 200 ticks), i.e. at least one of the sources has not been detected.

First, we varied the number of nodes to show the scalability of our approach. The number of msgs and reads growed linearly with the number of sensors in the system, while the number of failures decreased (see Table I).

Next, we evaluated the system in presence of different noise levels. When noise is lower than 4% the number of failures decreased, i.e. low noise increments the exploration, and also involved an increment in the number of msgs and reads (see Table II). Even with a noise factor equal to 10% the system was able to reach the convergence, but increasing the number of reads and msgs. We consider that the algorithm presents failure tolerance because it was able to find the diffuse event sources with a probability of 75%

Table I
Varying Sensor Number without Noise

Sensor Number	Reads	Msgs	Failures	Adj. avg.
500	311.30	518.82	35.5%	9.2
1000	397.38	651.64	15.2%	18.76
2000	681.67	1115.48	5.8%	37.09

Table II
Varying the noise factor γ

γ	Reads	Msgs	Failures
0%	397.38	651.64	15.2%
±2%	547.70	907.64	13.4%
±4%	698.31	1160.37	15.1%
±6%	776.21	1291.31	18.6%
±10%	878.73	1461.43	25.1%

using an average of 800 reads and 1500 msgs in a scenario with 1000 sensors.

The exploration level can be increased by changing T_w and P_a. Increasing the exploration rate, the number of failures decreases. The price is an increment in cost (msgs and reads). This trade-off between the quality of the results and the cost can be used in order to control the priority of the search process.

IV. Conclusions

In this paper we have proposed a new approach, based on a Mobile Multi-Agent technology, to detect diffuse event sources in dynamic and noisy environments using a wireless sensor network infrastructure. Our approach proposes a distributed and decentralized algorithm based on local interactions and local knowledge of the environment. Experiments conducted showed that the system is robust in front of sensor failures, when the system has enough number of sensors.

We plan to extend the experiments to evaluate the performance of our approach when sensors fail. Preliminary results show that mobile agents are able to find new routes, to reach the diffuse event sources even when sensors may fail. An extended version of this work can be found in [4].

References

[1] D. Blatt and A. O. Hero, "Energy based sensor network source localization via projection onto convex sets (POCS)," *IEEE Transactions on Signal Processing*, pp. 3614–3619, 2006.

[2] J. Weimer, B. Sinopoli, and B. H. Krogh, "Multiple source detection and localization in advection-diffusion processes using wireless sensor networks," in *30th IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2009, pp. 333–342.

[3] R. M. Ruair and M. Keane, "An energy-efficient, multi-agent sensor network for detecting diffuse events," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007, pp. 1390–1395.

[4] J. L. Fernandez-Marquez, G. Serugendo, and J. Arcos, "A decentralized approach for detecting dynamically changing diffuse event sources in noisy wsn environments," School of Computer Science and Information Systems, Birkbeck, University of London, Tech. Rep. BBKCS-10-06, Jun 2010.