

# Distributed event localization and tracking with wireless sensors<sup>\*</sup>

Markus Wälchli<sup>1</sup>, Piotr Skoczylas<sup>2</sup>, Michael Meer<sup>1</sup>, and Torsten Braun<sup>1</sup>

<sup>1</sup> Institute of Computer Science and Applied Mathematics  
University of Bern - Switzerland

<sup>2</sup> School of Computer and Communication Sciences  
EPFL - Switzerland

**Abstract.** In this paper we present the distributed event localization and tracking algorithm DELTA that solely depends on light measurements. Based on this information and the positions of the sensors, DELTA is able to track a moving person equipped with a flashlight by dynamically building groups and electing well located nodes as group leaders. Moreover, DELTA supports object localization. The gathered data is sent to a monitoring entity in a fixed network which can apply pattern recognition techniques to determine the legitimacy of the moving person. DELTA enables object tracking with minimal constraints on both sensor hardware and the moving object. We show the feasibility of the algorithm running on the limited hardware of an existing sensor platform.

**Keywords** Sensor network, monitoring, tracking, hardware

## 1 Introduction

Composed of hundreds or thousands of tiny battery-powered devices, equipped with an array of sensors and a wireless radio to communicate, sensor networks are utilized to monitor and interact with the environment. The target application scenario we aim at is terrain observation during night. By the help of a sensor network a certain area (e.g. public building) is observed and exceptional behavior should be detected. DELTA detects and tracks light sources (typically from flashlights) and sends the data to a management station in a fixed network, where pattern recognition techniques can be applied to determine whether the person is present legitimately or not. The basic operations are depicted in Fig. 1. To be able to identify movement patterns online, the sensor network has to provide meaningful data in real time.

DELTA is a fully distributed object tracking framework avoiding heavy data load to the base station. Furthermore, DELTA uses the sensor light measurements in decisions and computations like leader election and localization. DELTA

---

<sup>\*</sup> The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

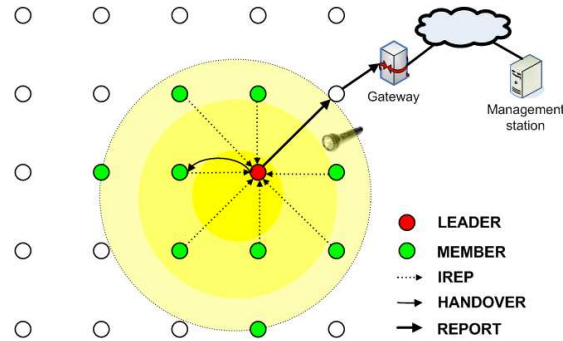


Fig. 1: DELTA observing an event and sending the data to to the management station for further processing.

maintains groups of nodes that are dynamically formed around objects as soon as these objects appear. A measurement-based election algorithm determines a group leader that is responsible for group maintenance, data gathering and processing, as well as reporting to the base station. The base station is connected to the Internet and may trigger alarm in case unusual behavior is detected. The alarm may be sent to any responsible authority. DELTA is designed to run on tiny nodes and has been implemented on the ESB sensor platform [1].

A requirement of DELTA is the knowledge of location information by the sensor nodes. This can be achieved by GPS or any other location service [2]. It could also be configured during deployment. The dependency on the node positions is no restriction as a terrain monitoring application intrinsically needs to know where the sensor nodes are placed.

In the next section we discuss related work in the areas of event localization and tracking. In section 3 we introduce the DELTA concepts and algorithms. The used hardware platform is described in section 4. Configuration data and the evaluation are provided in section 5. The paper ends with conclusions and outlook on future work in section 6.

## 2 Related work

Existing event monitoring applications can mainly be divided in two categories: event detection and event tracking. While event localization is an intrinsic part of many detection applications, it is only rarely considered in tracking applications. With DELTA we aim at filling that gap as the recognition of the movement pattern of a person needs precise and detailed information.

[3] uses the sensor nodes only to collect data. That data is routed to a base station equipped with more computational resources where the event location is computed. Avoiding the computational limitations of sensor nodes, the algorithm accepts heavy network load, in particular in the direction of the base station. Other approaches [4], [5], [6] maintain predefined and static clusters of sensor

nodes to localize appearing events. Creation and maintenance of these clusters leads to communication overhead, especially in sensor networks with temporal disconnections and failures of nodes. Sextant [7] is a distributed approach using Bézier regions to represent the locations of nodes and events. Sextant disseminates network properties in a restricted area causing too much delay for real time tracking.

EnviroTrack [8], [9] is a distributed event tracking algorithm. It supports event detection and tracking, but not localization. A moving object is tracked by dynamically established groups of nodes. DELTA performs a similar set of basic operations like EnviroTrack adding beneficial features like the usage of the sensor readings and event localization. The SensIt project [10], [6] considers localization, tracking and classification algorithms. In [10] signal energy based multilateration similar to the approach we propose was mentioned, but the algorithm was neither implemented there nor in later publications. The SensIt project is tailored to powerful sensor nodes like PDAs making their set of algorithms not really applicable in our context.

Some existing localization algorithms [3], [11] depend on the possibility to distinguish two kinds of signals transmitted by an event. The distance of the event is derived from the time difference of arrival (TDOA) of two different signals. For example, [3] uses the time difference of arrival between the shock wave and the muzzle blast generated by a gun to estimate the distance to a sniper. In many cases the dependency on two different kinds of signals is restrictive and not easy to fulfill. In contrast, we propose a localization algorithm that uses a multilateration scheme that solely depends on the values of the sensor readings of the nodes observing an event [12].

### 3 Architecture of the DELTA framework

A key problem of event tracking is the overhead of identifying and organizing the event relevant sensor nodes in a distributed manner with as little communication overhead as possible while providing a satisfactory degree of accuracy. Considering the application aimed at, persons are assumed to move quickly and the cones of light their flashlights produce may fluctuate even more.

To deal with these frequently changing conditions we use the sensor light readings to establish tracking groups fast and efficiently. Furthermore, the measurements are used to estimate the location of the moving object (light source). Details are explained in the next section. A limitation of existing tracking algorithms we overcome with DELTA is the dependency on the assumption that the communication range CR of the sensor nodes is significantly higher than their sensing range SR. In DELTA we ease this restriction with a 'passive heartbeat' mechanism: The leader periodically broadcasts heartbeats. To be able to estimate the location of a moving event, the group leader needs the light measurements and positions of its neighbors. Accordingly, the nodes that overhear the heartbeat respond their data. This message is overheard by the two hop neighbors which are thus inherently informed about the existence of a leader.

If needed, the presence of the leader can be disseminated even deeper into the network by the usage of passive heartbeats. Smart algorithms to distribute that information remain for future work.

### 3.1 State diagram of DELTA nodes

As DELTA is a distributed approach, the sensor nodes collaborate to compute the moving object location before sending the data to the base station. To achieve this different roles are assigned to the nodes. The states and state changes of the individual nodes and their roles are depicted in Fig. 2. The dashed lines show state changes caused by event observations. The solid lines show actions caused by messages from other nodes.

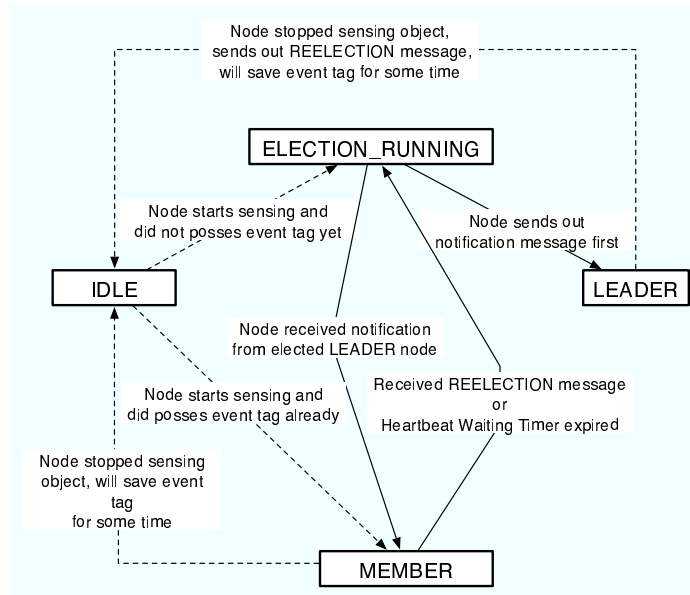


Fig. 2: State diagram of a DELTA node's roles.

One sensor node is the leader of a tracking group. The leader is responsible for maintaining group coherence, localization of the target position, and communication with the base station. All direct neighbors of the leader are group members and deliver their relevant tracking and localization data to the leader. All other sensors are idle. They check their sensors frequently to detect a target at appearance. Additionally, by overhearing the communication of other nodes they are informed about a target moving into their direction. In DELTA, the roles are assigned dynamically. If an object is sensed, but no leader has been established yet the election running state is entered.

### 3.2 Distributed leader election and group maintenance

Initially, all DELTA nodes are in the IDLE state. As soon as a target has been observed by a node, it switches to election running mode and schedules a timer according to the intensity of its light measurement. When the timer expires a heartbeat message is broadcast to inform about the presence of the group leader. The calculation of the timer is crucial as it determines the leader node. It partly depends on the hardware platform and discussed in more detail there (see section 4). Additionally, the leader node sets several state variables concerning the newly formed group: An event tag to identify the tracked event. It is used to announce the tracking group to the base station as well as to maintain group coherence. A round number which is increased whenever the leader broadcasts heartbeat message. The round number is used to check the freshness of the messages, i.e. whether they are relevant or whether they can be ignored. Furthermore, there is a TTL field defining the depth, the leader information has to be disseminated into the network. Finally, the leader node is responsible to ensure a controlled handover of the leadership once the observation of the moving object fails. The leader will then immediately broadcast a leader reelection message and switch to IDLE state.

Group members wait for periodic heartbeat messages to be informed about the presence of the leader. If DELTA applications have larger sensing ranges than communication ranges, not every node that senses the moving target is a direct neighbor of the leader. The information response messages to the leader as well as passive heartbeats are used to inform such nodes about the tracking group leader. Otherwise, concurrent tracking groups may evolve leading to confusion and message overhead in the direction of the base station (multiple leaders that report). The message flow with which DELTA overcomes the restriction  $\frac{SR}{CR} < 1$  or even  $\frac{SR}{CR} < \frac{1}{2}$  is depicted in Fig. 3.

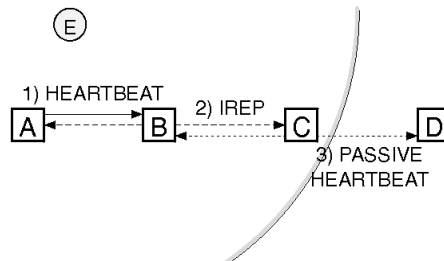


Fig. 3: Group communication in a DELTA network with  $\frac{SR}{CR} > 1$  and the TTL of the heartbeat messages from the leader set to 3.

As mentioned above, a leader sends periodic heartbeat messages to inform its neighboring members about its existence and to request the needed information from them. The neighboring members answer with IREP messages containing

that information. These messages are moreover overheard by all two-hop neighbors of the leader which are thus informed about the existence of the leader. Passive heartbeat messages are then retransmitted until the TTL value expires. Implementation details concerning medium access, communication pattern, and light measurements are provided in section 4.

The leader election process aims at quickly determining a single leader node which is able to cover a moving target reliably. Reliability includes several aspects: The leader should be able to keep its leading state as long as possible, thus minimizing the number of reelections and handovers. Therefore, its location should be close to the target position or the path the target is moving at. Furthermore, the leader should have enough battery power left to be able to bear the burden of increased communication and computation compared to normal group members. The election process needs to be fast to avoid periods when no leader is present. In contrast to EnviroTrack, the leader election delay of DELTA is deterministic (see section 4).

## 4 Hardware Platform and Implementation Details

As sensor platform the ESB sensor boards [1] are used. These nodes consist of a chip with a TI MSP430 microcontroller, 2kB of RAM, 60kB flash memory, and a low power consuming radio transceiver (868MHz) operating at a transmission rate of 19.2kb/s by default. Furthermore, the sensor boards are equipped with a number of sensors like luminosity, temperature, vibration, etc. The boards have mainly two restrictions: the comparatively low transmission rate and the resource limitations of the memory and the processing unit. This is mainly caused by the miniaturization of the implemented hardware. The sensors have to work with at most 3V DC and should consume as little energy as possible.

All experiments are based on the light sensor. The provided light measurement software was reimplemented as it allows only binary decisions (light on/off), what is not appropriate for our purpose. The light sensor (TSL245) is associated to a interrupt-capable register and an interrupt is generated on each positive edge of the output frequency of the TSL245 (see Fig. 4). For each interrupt a counter is incremented counting that frequency. This solution implies high costs in case of high irradiance. Therefore, the spectrum is limited to a frequency of 100kHz. All above is just considered as maximal bright. The output frequency of the TSL245 in a standard office on the desk during day is around 2kHz.

To detect moving light sources an exponentially weighted moving average filter has been implemented with  $\bar{x}_k = \alpha\bar{x}_{k-1} + (1-\alpha)x_k$ . The calculation of the mean  $\bar{x}_k$  thus only requires the storage of the past value of  $\bar{x}_{k-1}$  and the actual light measurement  $x_k$ . A light irradiance change is considered to be significant if the currently measured value differs more than a configurable threshold T from the average. Currently, T is set to 50. The advantage of having a moving average is the adaptivity to changing brightness in the environment. The moving average converges to the actual brightness, avoiding the permanent throwing of events

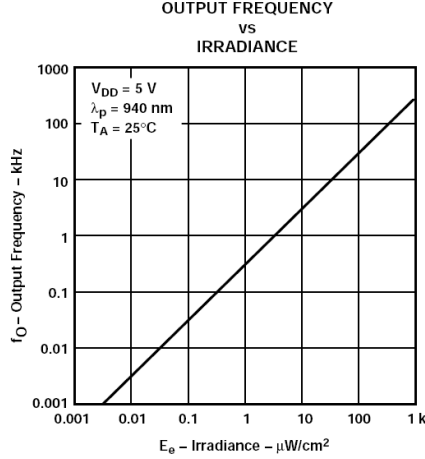


Fig. 4: TAOS TSL245 infrared to frequency converter. The output frequency of the TSL245 is linear to the measured light irradiance. The figure is from the TSL245 data sheet.

during day, building works, etc. For the current evaluation, the value of  $\alpha$  is set to 0.9.

As mentioned above, the computation of the leader election timer is crucial for the performance of DELTA. On the ESB platform we calculate the light irradiance every 200ms for exactly 100ms. As we limit the TSL245 output frequency to 100kHz, we get light values from a spectrum between 0 and 10<sup>3</sup>000. Nodes with high irradiance should compute short delays whereas nodes with low irradiance should compute long delays. The delay is computed as follows:

$$\Delta t [ms] = \frac{I_{MAX} - I_C}{10}$$

$$\Delta round [ms] = round(i) \cdot SAMPLE\_FREQUENCY$$

$$\Delta t = \begin{cases} \Delta t & , \Delta t < \Delta round \\ \Delta t = \Delta t - \Delta round & , \text{else} \end{cases}$$

$I_C$  is the currently measured irradiance.  $I_{MAX}$  is the maximum value of 10<sup>3</sup>000. Accordingly,  $\Delta t$  generates a delay between zero and one second. The SAMPLE\_FREQUENCY is the light measurement frequency of 200ms. The round variable is set to 0 when the election is initialized and then incremented each time the light value is measured (all 200ms). The computation of the delay allows the filtering of non-continuous irradiance peaks as long as the value is not too high, i.e. the timer does not expire before the next light measurement is done.

The ESB sensor boards include a TR1001 radio module. The provided software runs the radio with 19.2kbps. For our purpose this bandwidth is too slow as it causes high collision probabilities in case of message bursts as with the heartbeat/IREP message flow of DELTA. We therefore changed the software to

run with ASK modulation and 76kbps. To ensure 1-hop communication while minimizing energy consumption we adjusted the radio power to have communication at 1.25 meters, while having no communication at the doubled distance. The radio power adjustment results are depicted in Fig. 5.

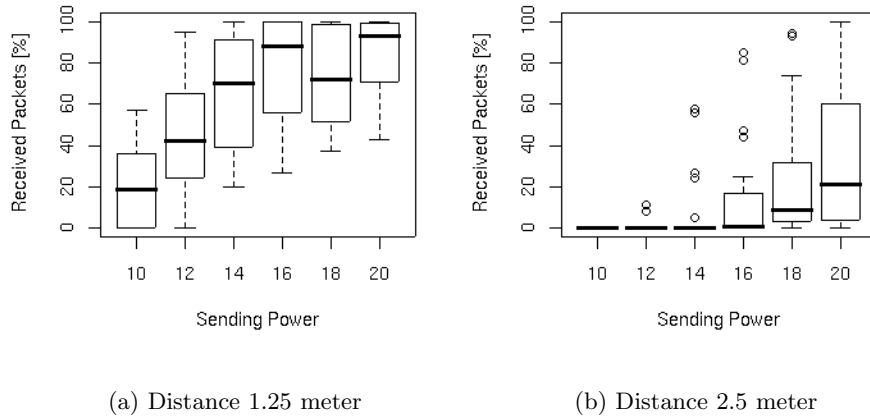


Fig. 5: Fraction of received messages for different transmission power levels.

We concluded that a sending power of 16 is the best choice according to the evaluation results. In dense networks the burst of IREP messages cannot be handled efficiently by CSMA with random backoff given a delay of 2 ms to switch from receive to transmit state and the approximately 14 ms to transmit a message. As the leader does not need all IREP messages in dense networks to calculate the target position we implemented the following mechanism: Within the heartbeat message the leader schedules  $n$  nodes from which it received IREP messages in previous rounds ( $n \leq 8$ ). Thus, the first  $n \cdot 14ms$  are reserved for the scheduled nodes to transmit their IREP messages. Not scheduled nodes send their IREP message after this time using common CSMA with random backoff. Obviously all nodes compete for the medium when a new leader is elected, as the leader has no neighbor information at that time.

## 5 Evaluation

For the evaluation EnviroTrack [8] was chosen as reference algorithm. The reason for this choice is the similarity of both concepts in distributed group establishment and maintenance. DELTA adds a number of features like light measurement based leader election and object localization.



## 5.1 Reference implementation of DELTA and EnviroTrack in the simulator

To compare DELTA to the original EnviroTrack algorithm as well as to support our real world experiments, we implemented both DELTA and EnviroTrack in the OMNeT++ network simulator [13].

The simulation settings from the original EnviroTrack evaluations have been taken to make our implementations and the original DELTA implementation comparable. The goal of their simulated network was to track T-72 battle tanks moving through an off-road environment. For the simulations a realistic object path neither with sharp turns nor following just a straight line was used. Only the tracking performance of both, DELTA and EnviroTrack have been evaluated. DELTA has been evaluated with a TTL of 1 (just heartbeats like EnviroTrack) and a TTL of 2. We vary the speed of the target object and the ratio between sensing range and communication range. All settings are repeated eight times and a 95% confidence interval is used. The sensor network consists of 160 nodes, arranged in a grid of 8 times 20.

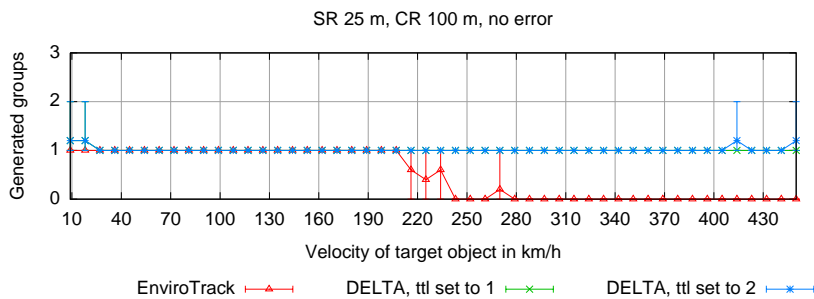


Fig. 6: DELTA and EnviroTrack with  $\frac{SR}{CR} = \frac{1}{4}$ . The average, minimum and maximum number of groups are shown.

In Fig. 6 results tailored to EnviroTrack are shown, i.e. the CR is significantly higher than the SR. Both algorithms have no problems in such scenarios. With this simulation we just show that DELTA performs not worse than EnviroTrack in a scenario tailored to EnviroTrack. In contrary, DELTA introduces shorter election delays (not depicted) and is therefore able to track objects at even higher speeds. Furthermore, DELTA performs equally well with the TTL set to one and two. This is obvious, as the ratio between SR and CR is too small to involve multiple groups. Consequently, the passive heartbeat mechanism does not provide any improvements here. The results are comparable to the original results from the EnviroTrack paper.

Fig. 7 shows what happens if the ratio between SR and CR becomes larger. In this scenario the SR is still smaller than the CR, but nevertheless the number of coexistent groups increases for both EnviroTrack and DELTA with TTL set

to 1. This shows that EnviroTrack is not suitable for dense sensor networks where the SR may become equal to or is even larger than the CR. With such network properties we are however confronted by the application we aim at. The decreasing number of leaders in EnviroTrack under higher speeds is due to the inability of EnviroTrack to build groups in time.

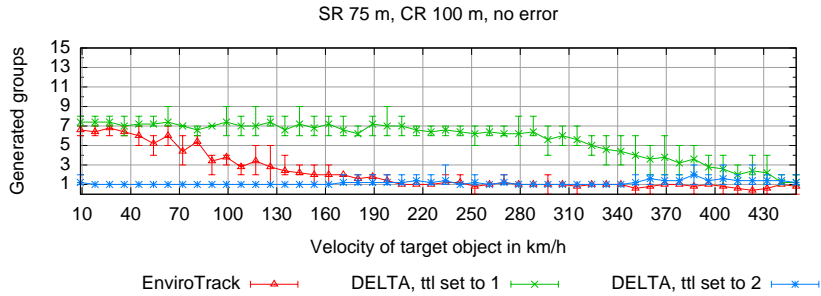


Fig. 7: DELTA and EnviroTrack with  $\frac{SR}{CR} = \frac{3}{4}$ . The average, minimum and maximum number of groups are shown.

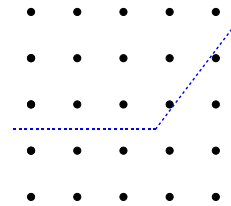
## 5.2 Real world performance of DELTA and EnviroTrack

Both DELTA and EnviroTrack have been implemented on the ESB sensor nodes. In the following the test setup and the results of the measurements are described.

**Tracking Performance of DELTA and EnviroTrack** The tests have been performed indoor in a completely dark room to avoid any external influences while tracking the moving light source. 25 nodes were placed in a 5x5 grid with a grid spacing of 1.25m. The transmission power has been reduced to 16 to



(a) Network setup



(b) Event path

just allow a node to communicate reliably with its grid neighbors. Two lamps,

common office equipment with a 25W bulb and a 40W bulb, have been used as light sources. The lamp was held about 1.5m over ground pointing 1.5m in front of the moving person. The directly illuminated area consisted of a circle with an approximately 4 meters diameter. The person walked through the sensor field at constant speed of about 0.2 m/s.

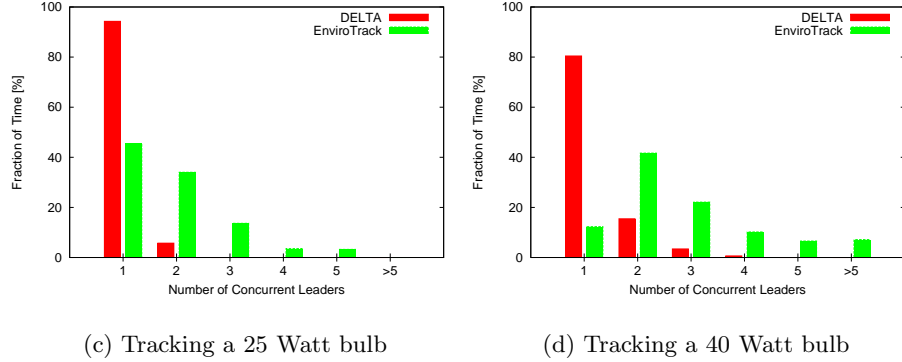


Fig. 8: Percentage of time a certain number of leaders are elected.

Fig. 8 shows the percentage of leaders concurrently elected for DELTA and EnviroTrack. DELTA has significantly less concurrent leaders. Multiple concurrent leaders result in multiple reports about the same object consuming energy and bandwidth. In particular a much higher load to the base station is the consequence what affects the overall network lifetime. With DELTA a higher message load in the tracking area is accepted, while the proportional overloaded paths to the base station are relieved. The amount of the message overhead of DELTA has not yet been considered and remains to be evaluated in future work. The on-demand establishment of the time slots for the IREP messages performs well. During all the simulations enough IREP messages were received at the leader to enable the computation of the event location. However, the localization remains to be finished in future work.

## 6 Conclusions and future work

The DELTA algorithm proposes a solution for object tracking and localization by combining features that are not yet available in other algorithms. Groups are dynamically created. DELTA works in smart dust environments with small radio ranges and proportionally high sensing ranges. The leader election procedure of DELTA is quick and very precise. It favors well suited candidates adaptively. The on-demand setup of the 'slotted' medium access for the IREP messages works

well. The results show that the usage of light measurements and IREP messages help to establish and maintain single groups to track the moving light source.

To distribute data in a larger area an optimized broadcast protocol could be used. A possible solution [12] could be implemented. Furthermore, it remains to be shown that measurement-based multilateration works on resource constraint sensor nodes. The light measurements are very exact, therefore we expect good results. The message overhead of DELTA remains to be investigated. Furthermore, we aim at deploying the algorithm in a realistic scenario to test its performance under real conditions. Finally, the occurrence of coexisting light sources and object classification remain to be investigated.

## References

1. Scatterweb: Sensor platform (2007) <http://www.scatterweb.net>.
2. Langendoen, K., Reijers, N.: Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks* **43**(4) (2003) 499–518
3. Simon, G., Balogh, G., Bap, G., Maróti, M., Kusy, B., Sallai, J., Lédeczi, A., Nádás, A., Frampton, K.: Sensor network-based countersniper system. In: *SenSys*, Baltimore, Maryland, USA (2004)
4. Zou, Y., Chakrabarty, K.: Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)* **3**(1) (2004) 61–91
5. Kumar, M., Schwiebert, L., Brockmeyer, M.: Efficient data aggregation middleware for wireless sensor networks. In: *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, Florida, USA (2004) 1579–1581
6. Brooks, R.R., Ramanathan, P., Sayeed, A.M.: Distributed target classification and tracking in sensor networks. *Proc. IEEE* **91**(8) (2003) 1163–1171
7. Guha, S., Murty, R.N., Sireer, E.G.: Sextant: A unified node and event localization framework using non-convex constraints. In: *MobiHoc'05*, Urbana-Champaign, Illinois, USA (2005) 205–216
8. Abdelzaher, T., Blum, B., Evans, D., George, J., George, S., Gu, L., He, T., Huang, C., Nagaraddi, P., Son, S., Sorokin, P., Stankovic, J., Wood, A.: Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In: *Proc. of 24th International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan (2004)
9. Luo, L., Abdelzaher, T.F., He, T., Stankovic, J.A.: Envirosuite: An environmentally immersive programming framework for sensor networks. *ACM Transaction on Embedded Computing System (TECS)* **V** (2006) 1–31
10. Li, D., Wong, K.D., Hu, Y.H., Sayeed, A.M.: Detection, classification and tracking of targets. *IEEE Signal Processing Magazine* **19**(2) (2002) 17–29
11. Capkun, S., Hamdi, M., Hubaux, J.P.: Gps-free positioning in mobile ad hoc networks. In: *Proceedings of HICSS*. (2001) 3481–3490
12. Wälchli, 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 ménuieres, France (2006)
13. Varga, A.: Omnet++ simulator (2006) <http://www.omnetpp.org/>.