Distributed Event Tracking and Classification in Wireless Sensor Networks

Markus Wälchli, Samuel Bissig, Michael Meer and Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern - Switzerland

***Abstract*— In this paper a distributed event localization, tracking, and classification framework (DELTA) is presented. An event is observed and tracked by dynamically established groups. Relevant sensor data is collected at dedicated nodes (group leaders) which are destined to perform all subsequent localization and group organization tasks. Based on the collected sensor data, both the event position and the signal strength(s) of the emitted signal(s) of the event are estimated. This enables DELTA to classify events based on the estimated signal emission power. Existing approaches either focus on accurate but cost-intensive collaborative signal processing (CSP) methods or on less accurate but more cost-efficient approaches mainly focusing on minimizing the communication load. DELTA bridges this gap by providing satisfying accuracy while keeping the network load at a reasonable level. The performance of the proposed framework is evaluated by simulation as well as by implementation on real hardware. In addition, problems of closed-form linear least square solutions for the localization task are discussed.**

Keywords Sensor networks, monitoring, tracking, signal processing, classification, tracking

I. Introduction

Composed of hundreds or thousands of tiny, battery-powered sensor nodes equipped with an array of sensors and a wireless radio to communicate, sensor networks are utilized to monitor and interact with the environment. A basic, but challenging task for many wireless sensor network applications is the detection, tracking, and classification of events.

In our previous paper [24], presented at the 5th International Conference on Wired/Wireless Internet Communications (WWIC) 2007, the focus was on the detection and tracking of events. In this paper a number of substantial increments are presented. The DELTA framework has been enhanced with the localization and classification logic, which bases on a well-known sensor model. Nonlinear and linearized solutions to the localization and classification problem are discussed. The associated related work has been included. Furthermore, the communication costs of the detection and tracking performance have been investigated.

To this date the classification of events is mainly done by applying cost-intensive CSP methods. On the other hand, many existing event detection and tracking algorithms do without accurate event positioning. Thus, the communication need can be kept comparatively low but the classification of different events is no longer possible. In contrast, DELTA addresses both tasks. Moreover, DELTA is designed for sensor networks consisting of small, resource-constraint, and error-prone nodes. DELTA uses the measurements of the event observations to both efficiently organize event tracking groups and accurately localize and classify events. The basic operations of DELTA are shown in Fig. 1.

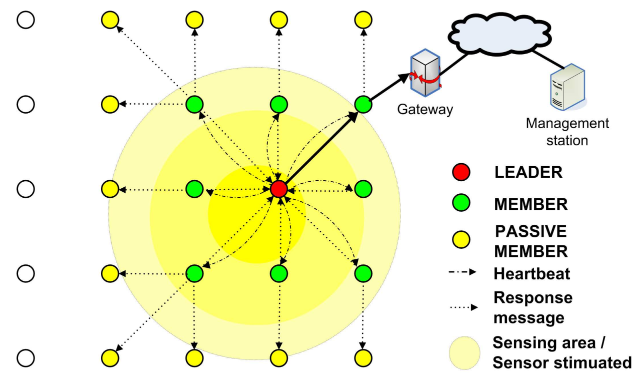


Fig. 1: Event detection, tracking group organization, localization, and reporting with DELTA.

A measurement-based leader election algorithm determines a unique group leader which is responsible for the group maintenance. Additionally, this approach facilitates in-network data gathering and processing on a dedicated node. Finally, the leader reports the tracking and localization results to a base station which is connected to the Internet where the data is stored and/or further processed. Based on the gathered information the leader is able to estimate both the location and the emitted signal power(s) of the event. There are two restrictions on the kind of signals which can be used in the localization and classification procedures. First, the computation of position and signal emission power require an attenuation model for each considered signal (e.g., sound, vibration, RSSI). Second, to be able to classify distinct events the signal emission power of specific events needs to be characteristic, i.e., more or less constant. Considering classification, the accurate event position is of less importance and mainly derived as byproduct in the emitted signal power computations. For other applications the event location might be of different interest, though.

For the current evaluation, the target application is terrain observation during night. DELTA detects, classifies, and tracks different light sources (typically from flashlights) and sends the event data, i.e. the computed event position and the signal magnitude to a management station in a fixed network, where the data is stored and clustering algorithms are applied to learn the different light sources.

DELTA is used to detect and track single events. There are no restrictions on the detection and tracking of multiple events as long as they occur in spatial sufficiently disjoint areas. If the event areas overlap, further statistical techniques might be necessary. Moreover, DELTA requires the the sensor nodes to know their location. This can be achieved by GPS or any other location service ([10], [18]). Considering static networks with

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.

a predefined topology (e.g., monitoring of stockrooms), the node positions might even be set before or while deployment.

The next section discusses related work. The DELTA detection and tracking concepts are introduced in section III . Different localization methods are presented in section IV. Section V describes the used hardware platform. Configuration data and the evaluation are provided in section VI. The paper ends with conclusions and future work in section VII.

II. Related work

Existing event monitoring applications can mainly be divided into two categories: On the one hand, there are contributions from the networking and communications research field focusing on efficiency and communication load minimization. These approaches mainly support event detection and tracking. The network is either divided into static monitoring areas or there are spatially-restricted dynamic tracking groups established. On the other hand there are contributions from the CSP research field. These works focus on localization and classification accuracy often taking high communication load into account.

A. Contributions Focusing on Networking

[26], [9], [2] focus on group formation. [26] divides the network in predefined and static clusters, which does, in general, not reflect the effective event occurrence topology and might lead to organization and communication overhead. In [9] the group is organized by a quorum-based consensus mechanism. The approach requires a multi-step negotiation procedure and is limited to applications where the sensing range of the event is smaller than half the communication range. In the work of [2], tracking groups are dynamically established according to the target (event) velocity. The group formation bases on a message-passing-like communication scheme. The group formation again requires rather high communication costs.

EnviroTrack [1], [14] is a distributed event tracking algorithm, supporting event detection and tracking, but not localization. A moving object is tracked by dynamically established groups of nodes. Group leaders are determined based on a random timer. Once elected they immediately start to organize their groups. The leader is responsible to report event relevant data to the base station, and to initialize hand-over in case the event leaves its tracking region. DELTA performs a similar set of basic operations as EnviroTrack, adding additional features such as the consideration of sensor measurements in the detection and tracking tasks and precise event localization and event classification.

Another approach organizing tracking groups has been proposed in [4]. In IDSQ, a group leader incrementally queries group members until a computed belief state is considered significant. The goal of IDSQ is to query as few sensor nodes as possible to still get a meaningful result. The main drawback of IDSQ is the incremental querying which disqualifies it as a solution for real-time tracking. Moreover, a mulitcast/broadcast querying might be more efficient.

Distributed approaches providing coarse-grained node localization have been proposed in ([6], [5]). Sextant [6] applies Bézier regions to represent the locations of nodes and events. To derive and update these regions, Sextant disseminates network properties (positive and negative connectivity constraints) in a restricted area. Drawbacks are high delays and a rather low localization accuracy. In [5] a similar approach using rectangles instead of Bézier regions is used.

B. Source Localization and Classification based on CSP

The localization of events based on energy decay models has a long tradition in the signal processing community. Accordingly, these models have been adapted to wireless sensor networks in a number of works ([13],[20],[11],[12],[21],[3],[17]). A common property of these approaches is their focus on the optimization of the localization accuracy. On the other hand, less focus is spent on network load and energy constraints. The discussed algorithms localize sound sources, but could be substituted by any other energy decay models (e.g. light, seismic, etc.).

Source localization and classification were extensively investigated in the SensIt project ([13],[20],[11],[12],[21]). The main focus of the project was on the localization of multiple, coexisting events. Therefore, statistical methods based on time series of event measurements were proposed. In [13] three different classification algorithms, namely k-NN, maximum likelihood (MA), and support vector machines (SVM), were investigated. Limitations of all statistical approaches are their rather centralized nature and their need for a considerable amount of data to provide statistically relevant results. Refinements of these statistical methods have been investigated in [20], [21]. Besides the multi-event localization, single event localization has been addressed in [11]. Four different nonlinear optimization methods for single event localization were considered: Exhaustive Search (ES), Multi-resolution search (MR), the Nelder and Mead simplex downhill algorithm (SD), and the conjugate gradient descent method (CG). To avoid local optimums, all algorithms search the feasible solution space by applying the respective optimization algorithm on each point of a grid overlaying the solution space. The authors have shown that the complexity for all but ES is about the same. They suggest to use MR, GD, or SD after having applied a coarse-grained ES to reduce the solution space. The computational burden of searching the solution space on tiny sensor nodes is too high, though.

In subsequent research [12], the nonlinear optimization has been replaced by a closed-form solution. For real time performance a linearized solution appears very attractive due to its simplicity and computational efficiency (see also [10],[22]). However, the linearization requires an over-determined system, else it lacks drastic accuracy. Redundant data may often not be available in sensor networks, though. In those situations a nonlinear solution might still provide useful information while the linearized methods fail.

In [3] the nonlinear localization of single and multiple events is investigated. The positioning of multiple events is based on the Levenberg-Marquart algorithm, which locally uses the Newton-Raphson method. Single event positioning problems are addressed by maximum-likelihood-based methods, requiring global knowledge. [17] applies a decentralized incremental subgradient optimization to localize an event. Thereby, a parameter estimate is circulated through the network and incrementally updated until a precision threshold is reached, or the maximum number of search steps is exceeded.

III. Event Detection and Tracking

A key problem of event detection and tracking is the complexity 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. In many tracking applications the location of the event occurrence might not be predictable. Moreover, depending on the emitted event amplitude a large event area could result. Also, the event might move fast, possibly even performing a sequence of successive turnarounds. Such properties are difficult to predict and challenge any generic event detection and tracking algorithm.

A central feature of the DELTA architecture, to deal with generic and frequently changing conditions, is the significance of the sensor measurements in the group establishment and maintenance tasks. Moreover, with DELTA the common assumption that the communication range (CR) of the sensor nodes is significantly higher than the sensing range (SR) is overcome: As soon as a leader evolves, it communicates its state to its neighborhood. This requires some periodic notification. Moreover, a periodic feedback message containing event information of the neighbor nodes is mandatory for the localization and classification of the event. These feedback messages are overheard by all two-hop neighbors of the leader, which are thus implicitly informed about the existence of the leader. If needed, the presence of the leader can be disseminated even deeper into the network by rebroadcasting passive heartbeats (see subsection III-B).

A. State diagram of DELTA nodes

To localize and track a moving event in a distributed manner some collaboration among the network nodes is needed. To achieve this, DELTA assigns different roles to the nodes. The states and state changes of the individual nodes and their roles are depicted in Fig. 2.

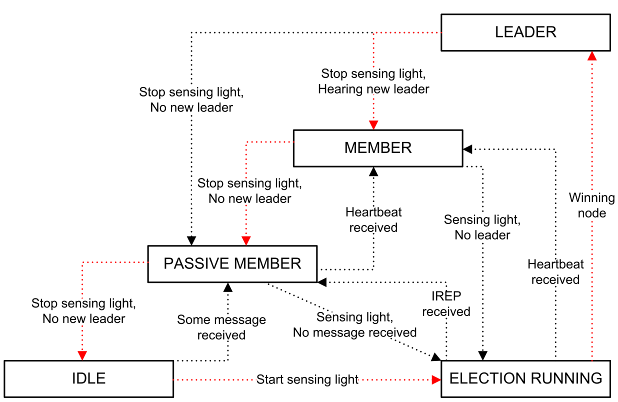


Fig. 2: State diagram of a DELTA node's roles.

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 either passive members or idle. The passive member state has been introduced to inform the neighborhood of an event tracking group about a possibly upcoming event. Moreover, confusion caused by state switches can be prohibited. In all states the sensor nodes periodically check their sensors to detect an event appearance. If there is no communication going on, but an event is sensed, all affected nodes enter the leader election state and compete for the leadership. In DELTA all roles are assigned dynamically.

B. Distributed leader election and group maintenance

Unless an event has been sensed, all DELTA nodes are in state IDLE. As soon as an event is observed by a node, it switches to ELECTION RUNNING state and schedules a timer according to the amplitude of the measurement, i.e., the stronger an event is sensed, the shorter the timer is set. When the timer expires a heartbeat message is broadcast to inform the neighborhood about the presence of the group leader. All receiving nodes immediately cancel their own timer and become a group member. The calculation of the timer is crucial as it determines the leader node. It partly depends on the used hardware and is, therefore, described in detail in section V.

The leader node initializes and maintains several variables concerning the newly formed group. To identify the observed event a temporary unique event tag is set. It is used to announce the tracking group to the base station as well as to maintain group coherence. To avoid the processing of outdated information a round number is used. It is increased whenever the leader broadcasts a heartbeat message. Thus messages with a round number smaller than or equal to the current round can be ignored. A TTL field defines the depth the leader information is disseminated into the network. The leader node is also responsible to ensure a controlled handover of the leadership once its observation of the moving event ends. The leader will then immediately broadcast a leader reelection message, optionally addressing the subsequent leader, and switch to IDLE state.

Considering DELTA applications with larger sensing ranges than communication ranges, not every node that senses a moving event is a direct neighbor of the leader. Accordingly, these nodes cannot be addressed by the heartbeat messages. However, the information response (IREP) messages, which report the location and classification relevant data of the group members, cover all nodes two hops away from the leader node. In case even larger sensing ranges are required, a passive heartbeat mechanism might be used to inform nodes farther away about the existence of an event. Of course, this implies some overhead. Optimized broadcasting techniques might be used ([7],[25]). In most cases the required heartbeat/IREP data exchange procedure should be sufficient to cover the whole event area, though. The message flow of DELTA overcomes the restriction $\frac{SR}{CR} < 1$ or even $\frac{SR}{CR} < \frac{1}{2}$ as illustrated in Fig. 3.

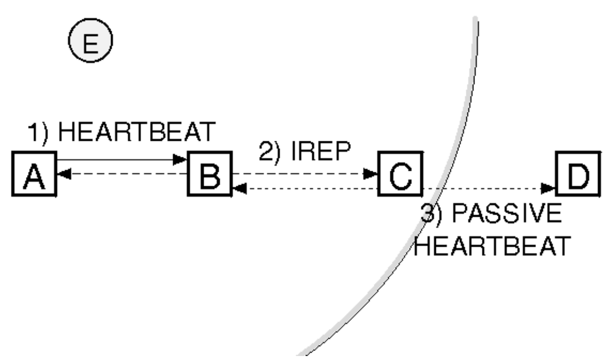


Fig. 3: Group communication in a DELTA network with $\frac{SR}{CR} > 1$ and the TTL of the heartbeat messages from the leader set to 3.

The avoidance of multiple, concurrently existing tracking groups is desirable, else network confusion and message overhead, in particular into the direction of the base station, might occur. The leader election process aims at quickly determining a single leader node which is able to cover a moving event reliably. Reliability includes several aspects: The leader should be able to keep its leading state as long as possible, minimizing the number of reelections and hand-overs. Consequently, its position should be close to the event location or the path the event is moving at. Furthermore, the leader must have enough battery power left to be able to bear the burden of temporary increased communication and computation load. Finally, 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 V), which increases the performance of DELTA.

IV. Event Localization and Classification

This section presents the localization and classification procedures. The group establishment and maintenance algorithm, introduced in the last section, provides the leader node with the relevant data to localize and classify an occurring event.

A. Measurement-based source localization

In order to estimate the location of events, an adequate sensor model is needed. Assuming that the emitted signal propagates isotropically (e.g. sound and light from point sources), the received signal ρ_i at a sensor node i located at position ξ_i is related to the event position $\mathbf{x}$ according to the model:

$$\rho_i = \frac{c}{\|\mathbf{x} - \xi_i\|^\alpha} + \omega \tag{1}$$

where c represents the amplitude of the emitted signal, α is the attenuation degree of the considered signal, ω is some additional white gaussian noise, and $\|.\|$ is the Euclidean norm.

In some existing approaches the ratio of the event measurements of pairs of sensors is used to compute the event location. Thus, the emitted amplitude c can be truncated. On the other hand, this adds the restriction that the denominator must not become zero. The equation considering two nodes i and j, the noise is considered by overdetermining the system, becomes:

$$\frac{\rho_i}{\rho_j} = \frac{\|\mathbf{x} - \xi_j\|^\alpha}{\|\mathbf{x} - \xi_i\|^\alpha} \tag{2}$$

For the classification of the events we aim at knowing the emitted signal strength and therefore consider the signal amplitude. Currently, we localize light sources. Accordingly, the attenuation coefficient α is equal to 2 and Eq. (1) can be rewritten as

$$\|\mathbf{x}\|^2 + \|\xi_i\|^2 - 2\mathbf{x}^T\xi_i - \frac{c}{\rho_i} = 0 \tag{3}$$

Given N sensors, N equations of the form (3) can be formulated. The quadratic constraints on the unknown variable $\mathbf{x}$ can be removed by subtracting the i = 1 equation from the rest (i $\neq$ 1), resulting in a system of N-1 linear equations of the form

$$2\left(\xi_1 - \xi_i\right)^T \mathbf{x} + c\left(\frac{1}{\rho_1} - \frac{1}{\rho_i}\right) = \|\xi_1\|^2 + \|\xi_i\|^2 \tag{4}$$

which can be solved with the closed-form standard linear least square (LLS) method $E = (A^TA)^{-1}A^Tb$, where A is a matrix containing the variables of the instances of Eq. (4) and b is a vector containing the constant parts of the instances of Eq. (4). As there are n = 3 unknown variables in Eq. (4), there are n+1 sensors needed to get a unique solution for the above system of equations. If Eq. (2) is used to build the system of linear equations, the inverse of the matrix in the LLS method might not be computable due to nodes having equal coordinates.

In the simulation part of this work we will show that the linearization lacks drastic accuracy if the linear system is not over-determined. Therefore, we reformulate Eq. 1 as a nonlinear least square objective function

$$f(\mathbf{x}, c) = \sum_{i=1}^{k} \left(\rho_i - \frac{c}{\|\mathbf{x} - \xi_i\|^\alpha}\right)^2 \tag{5}$$

which can be minimized using nonlinear optimization methods. For DELTA we evaluate two simple optimization methods, namely Nelder-Mead's Simplex Downhill (SD) [15],[16] algorithm and the Conjugate Gradient descent method (CG) [16]. Both algorithms are not protected against finding local minima. Accordingly, the determination of a well-placed starting point, respectively simplex, is crucial. Finding the global minimum is a challenging problem. Moreover, it is very cost-intensive and therefore not suitable for our purposes, i.e., it needs an additional search procedure (e.g., Monte Carlo) what makes it unfeasible to be run on the sensor nodes.

1) Simplex Downhill: The simplex downhill algorithm requires only function evaluations. A simplex is a geometrical figure that consists of N + 1 points in N dimensions. In two dimensions, a simplex is a triangle, in three dimensions it is a tetrahedron, and so on. The simplex downhill method starts with an initial simplex, the location of which is crucial for the performance of the algorithm. Then a sequence of geometrical operations (reflection, expansion or contraction) are applied on the simplex always aiming at minimizing it, i.e., determining the highest point and transform it to a lower point. The termination criteria is met when the vector distance in a step is below a certain threshold.

2) Conjugate Gradient Descent: At a given N-dimensional point $\mathbf{P}$, not only $f(\mathbf{P})$, but also the gradient $\nabla f(\mathbf{P})$ must be computable. The gradient $\nabla f(\mathbf{P})$ is a vector field that points into the direction of the largest increase of $f(\mathbf{P})$. In its simplest form, the minimization is in the direction of the local downhill gradient $-\nabla f(\mathbf{P})$ (Steepest Descent). In many cases however, the Steepest Descent method needs many steps to terminate. Therefore, the conjugate gradient procedure was proposed, which operates similarly as Steepest Descent. Thereby, the direction of the descent is computed slightly different, requiring fewer steps to terminate.

V. Hardware Platform and Implementation Details

The ESB sensor boards [19] are used for the experimental evaluation. 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 such as 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 basically 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 TSL245 light sensors [8]. The provided light measurement software was re-implemented as it allows only binary decisions (light on/off), which is not appropriate for our purpose. The light sensor is associated to a interrupt-capable register. An interrupt is thrown on each positive edge of the output frequency of the TSL245

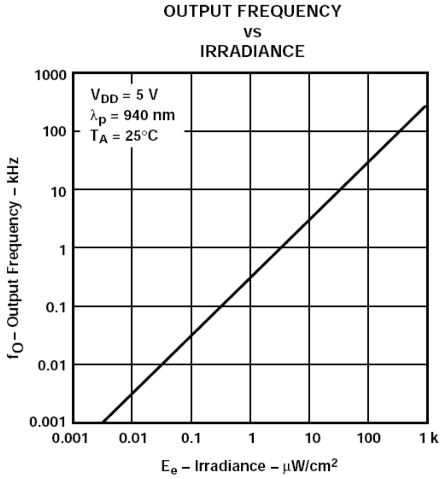


Fig. 4: Output of the TAOS TSL245 infrared to frequency converter [8].

(see Fig. 4). For each interrupt, a counter is incremented. 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 maximum brightness. The output frequency of the TSL245 in a standard office on the desk during day is around 2kHz.

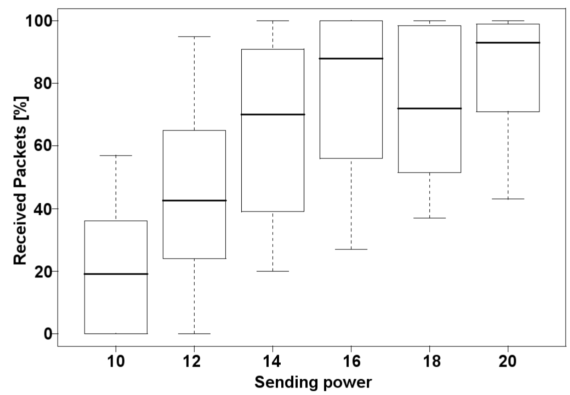
To detect moving light sources, an exponentially weighted moving average filter has been implemented with $\overline{x}_k = \alpha\overline{x}_{k-1} + (1-\alpha)x_k$. The calculation of the mean $\overline{x}_k$ thus only requires the storage of the past value $\overline{x}_{k-1}$ and the actual light measurement x_k. A light irradiance change is considered as 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 filter is the adaptivity to changing brightness in the environment. The moving average filter converges to the actual brightness, avoiding permanent throwing of events during day, building works, etc. In the current application, the value of α is 0.9.

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

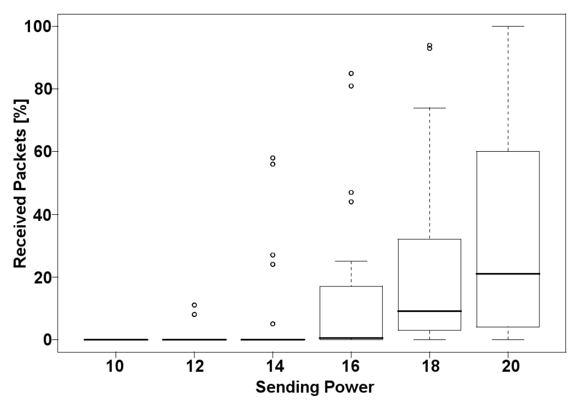
$$\begin{aligned}
\Delta t[ms] &= \frac{I_{MAX} - I_C}{10} \\
\Delta round[ms] &= round(i) \cdot SAMPLE_FREQUENCY \\
\Delta t &= \begin{cases} \Delta t & , \Delta t < \Delta round \\ \Delta t = \Delta t - \Delta round & , \text{else} \end{cases}
\end{aligned}$$

I_C is the currently measured irradiance. I_{MAX} is the maximum value of 10'000. Accordingly, Δt 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 (every 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 has been done.

The ESB sensor boards have a TR1001 radio module implemented. The provided software runs the radio with 19.2kbps. For our purpose this bandwidth is too small, as it causes high collision probabilities in case of message bursts, e.g., with the heartbeat/IREP message flow of DELTA. The software was therefore changed to run with ASK modulation and 76kbps, which consumes slightly more energy. Sending with maximum power consumes too much energy. Therefore, the sending power needs to be adjusted so that the communication between every neighboring pair of nodes is highly probable, whereas the communication between nodes which are two hops away from each other is improbable. Neighboring nodes are placed 1.25 meters away from each other. Accordingly, the sending power was adjusted to cover a range of approximately 1.75 meters. The results of the sending power control evaluation are shown in Fig. 5.



(a) Distance 1.25 meters



(b) Distance 2.5 meters

Fig. 5: Fraction of received messages for varying sending power.

The maximum sending power of the TR1001 is 99. From the evaluation we concluded that a sending power of 16 is the best choice for the current network settings. At a distance of 1.25 meters a high fraction of packets is received, while at a distance of 2.5 meters only few packets are received. Setting the sending power to a lower level involves too much packet loss at 1.25 meters, whereas a higher level involves a too high receive fraction at 2.5 meters.

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. On the other hand, the leader requires only a limited number of IREP messages to compute the event position. Therefore, we implemented an on-demand slotting mechanism: Within the heartbeat message the leader schedules at most n, with $n \leq 8$, members. The leader learns those members from IREP messages. In all subsequent communication all addressed members respond in the first $n \cdot 14$ ms according to their position in the schedule. All not scheduled members send their IREP message after this time using common CSMA with random backoff. Obviously, all nodes compete for the medium when a new leader has been elected as the leader has no neighborhood information at that time.

VI. Evaluation

The evaluation is divided into two parts. In the first part, the detection and tracking performance of DELTA, in comparison to EnviroTrack [1], is investigated. The choice of EnviroTrack is due to the similarity of both concepts in distributed group establishment and maintenance. In the second part, the performance of the different localization approaches is shown. The outcome of the localization procedure, i.e., in particular the amplitude estimates, constitute the basis for any subsequent classification.

A. Detection and Tracking performance of DELTA

To simplify a comparison between DELTA and the original EnviroTrack algorithm, both DELTA and EnviroTrack have been implemented on the ESB sensor boards as well as in the OMNeT++ network simulator [23].

1) Simulated Performance: The simulation settings from the original EnviroTrack evaluations have been taken. The goal 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. Just the detection and tracking performance were evaluated. DELTA has been evaluated with a TTL of 1 (just heartbeats like EnviroTrack) and a TTL of 2 (reporting event relevant data and informing the two-hop neighborhood about a leader). The speed of the target object and the ratio between sensing range (SR) and communication range (CR) varied. All settings have been repeated eight times and a 95% confidence interval was used. The sensor network consists of 160 nodes arranged in a 8 x 20 grid. The distance between any two neighbors is 100 meters.

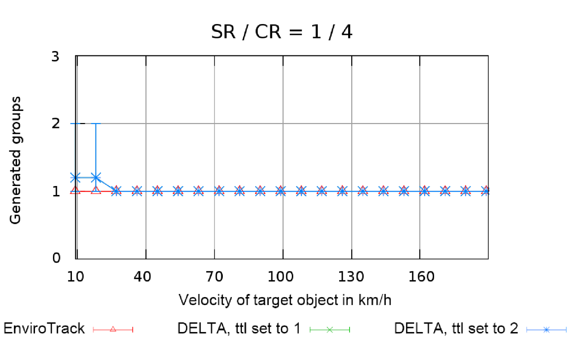


Fig. 6: Average, minimum and maximum number of groups with $\frac{SR}{CR} = \frac{1}{4}$.

Fig. 6 shows results with the CR being significantly higher than the SR. Such scenarios are tailored to EnviroTrack and and both protocols perform equally well. DELTA performs equally well with the TTL set to 1 or 2. However, this is not surprising considering the ratio between SR and CR of $\frac{1}{4}$. In such scenarios, groups can easily be organized only by the heartbeat mechanism. Though, if only using heartbeat messages no localization and classification of the events is possible.

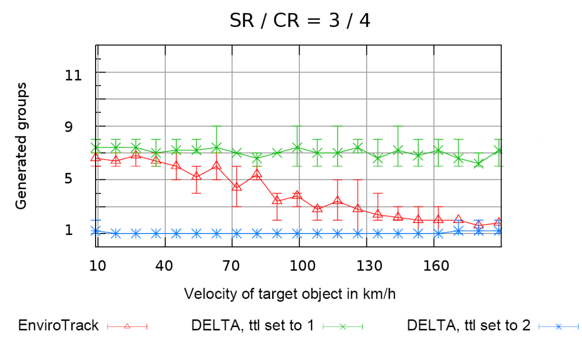


Fig. 7: Average, minimum and maximum number of groups with $\frac{SR}{CR} = \frac{3}{4}$.

Fig. 7 shows performance results if the restriction of the $\frac{SR}{CR}$ ratio being smaller than $\frac{1}{2}$ is overcome. Even when considering a ratio of $\frac{3}{4}$, which only slightly hurts the above condition, the number of coexistent groups increases considerably for both EnvrioTrack, and DELTA with the TTL set to 1. This shows that in scenarios with higher SRs a passive heartbeat mechanism alone is not sufficient. Enhancing the heartbeat procedure with the IREP messages solves the problem of concurrent leaders and supplies the leader with the information needed to support localization and classification. The decreasing number of leaders in EnviroTrack for higher speeds is due to the inability of EnviroTrack to build groups in time.

2) Performance in Real-World Experiment: All tests have been performed indoor in a shaded room to minimize external influences. 25 nodes have been arranged in a 5x5 grid with a spacing of 1.25 meters. The setup is depicted in Fig. 8.

Fig. 8: Experiment setup with 25 sensor boards.

The transmission power was reduced to 16 to restrict communication to grid neighbors only. 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 above ground pointing to floor 1.5m in front of the moving person. The directly illuminated area was a circle with a diameter of approximately two meters (25W bulb), respectively four meters (40W bulb). The person covered a distance of about seven meters, walking at a constant speed of about 0.3 m/s. The person walked along a straight line through the sensor network (illustrated in Fig. 9). Each experiment was repeated five times and a 95% confidence interval was used.

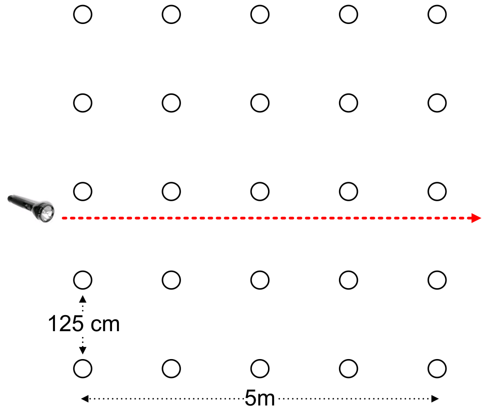
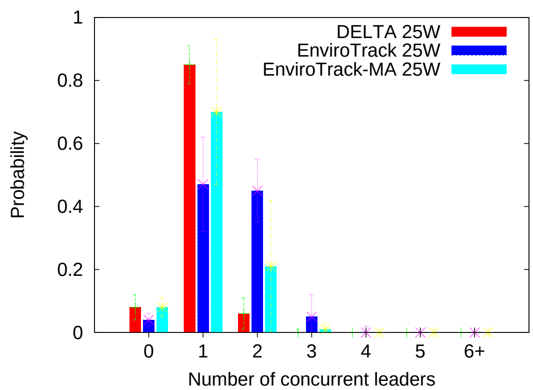
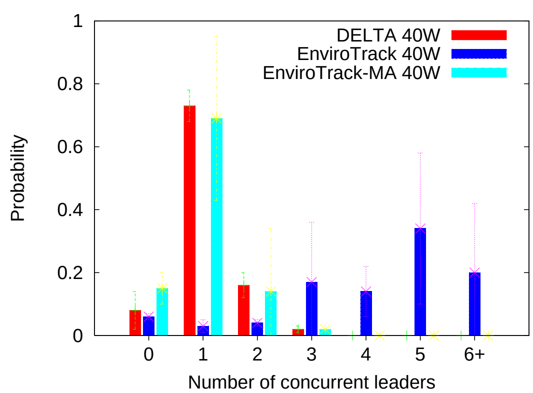


Fig. 9: Event path through the sensor network.

To see the impact of the different enhancements of DELTA, a second EnviroTrack version (EnviroTrack-MA) enhanced with the moving average filter has been implemented.



(a) Tracking of a 25W bulb



(b) Tracking of a 40W bulb

Fig. 10: Fraction of concurrent leaders.

The results of the detection and tracking performance of DETLA and EnviroTrack are shown in Fig. 10. When the sensing range increases (40W bulb), DELTA produces significantly fewer concurrent leaders than the original EnviroTrack implementation. This supports the simulation results. Concurrent leaders produce unnecessary event reports, producing confusion while wasting energy and bandwidth. The network load towards the base station is increased, affecting the overall network lifetime.

The performance of EnviroTrack enhanced with the moving average filter is nearly as good as with DELTA. The fast convergence of the MA filter at the border of the sensing area suppresses many nodes close to that border from being elected. However, EnviroTrack still has the drawback that neither localization nor classification is possible. Moreover, there is a slightly higher fraction of time without any leader. The increased number of state switches caused by the moving average filter in combination with the additional internal states of EnviroTrack lead to this behavior. The communication costs of DELTA and EnviroTrack are indicated in Fig. 11.

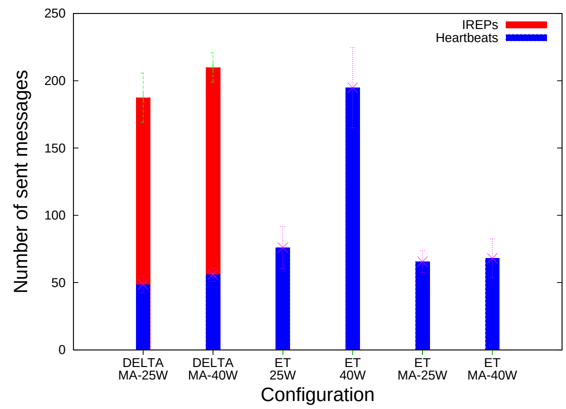


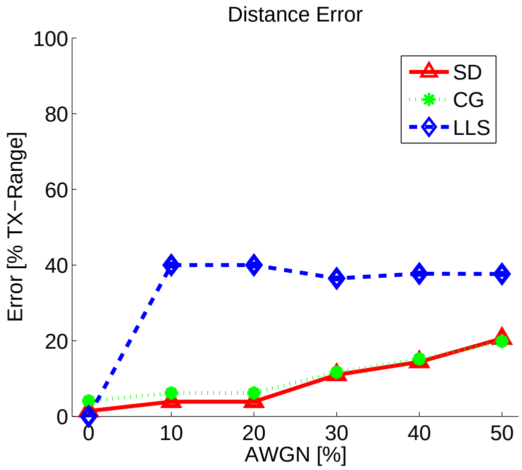
Fig. 11: Number of sent messages of the different approaches.

In order to make the localization and classification of the event, the reception of 3 IREP messages is required. The number of sent IREP messages can be restricted by the number of assigned slots. As soon as all slots are assigned, only the addressed members are allowed to send their IREP message. Theoretically, setting the number of slots to 3 was sufficient. However, due to packet loss the current implementation on the ESB sensor boards required 5 slots to receive the needed 3 IREPs. This value depends on the network structure and the used hardware, though. Fig. 11 shows that for a higher SR the communication costs of DELTA are similar to those of EnviroTrack while inherently providing the information needed for the localization and classification. EnviroTrack enhanced with the MA-filter was able to keep the number or sent messages small. Obviously, no localization and classification is possible with that approach. Though, if this is not required, a heartbeat-based approach enhanced with a MA-filter might be sufficient.

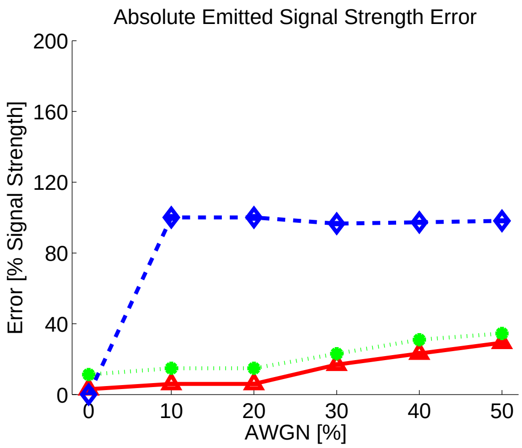
B. Localization Accuracy

DELTA provides the leader node with the information needed to localize and classify an event. In a first step, different possible localization methods have been evaluated in Matlab. The SD and CG methods together with a closed-form linearized least square (LLS) solution have been considered (see section IV).

1) Simulation of Localization Performance: For the evaluation four nodes were arranged in a square with a side length of 125 cm. An event was placed randomly within this square. The localization was performed 200 times with a confidence of 95%. Both, SD and CG require well located starting points. For SD the simplex is located at the center of area of the sensing nodes and their measurements. For CG the center of area only is sufficient. Noise of the sensor measurements is modeled as additional white gaussian noise (AWGN). The noise level has been increased from zero to 50%, in steps of 10%. The results are shown in Fig. 12.

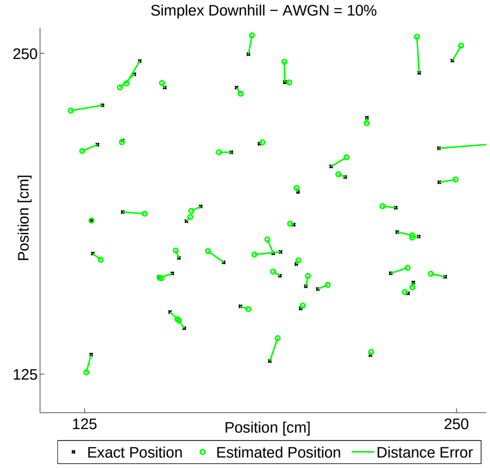


(a) Distance Error

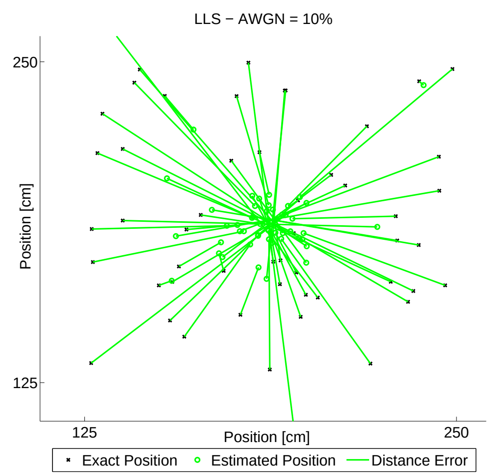


(b) Signal Strength Error

Fig. 12: Accuracy of LLS, SD, and CG.



(a) Distance Error with SD



(b) Distance Error with LLS

Fig. 13: Location estimation accuracy of SD and LLS.

The results show that the LLS method does neither work satisfactorily considering the computation of the position of the event nor its emitted signal amplitude. Almost independently from the noise level, the position error is always about 40% of the transmission range, i.e., the grid length in this scenario. The signal amplitude error is even worse. The problem of the LLS method is illustrated in Fig. 13. To improve readability, only 50 out of 200 estimations are depicted.

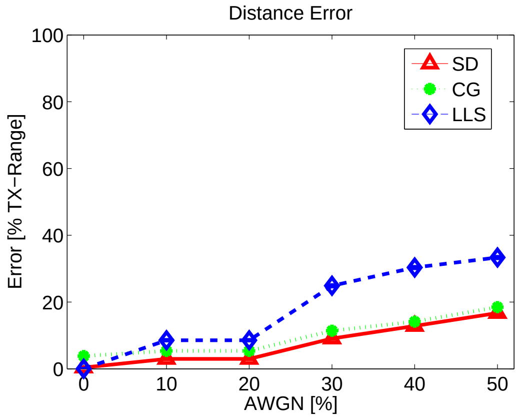
Only little affected by the noise level, the majority of the LLS estimations is close to the center of the sensing area. The distance errors (lines between the exact event positions and their estimations in Fig. 13) are accordingly high. The accuracy of the LLS method is improved if the system is overdetermined, i.e., if more than four sensor nodes are used in the scenario above. This implies more communication load. Moreover, the probability of receiving the needed information is decreased (see section VI-A.2). In Fig. 14 results with 6 sensing nodes are shown. The two additional nodes are placed at the positions (175,125) and (175,250).

The performance of the LLS method is better in an overdetermined system, though it does still not reach the performance of the nonlinear methods. In conclusion, both SD and CG outperform the LLS method in all scenarios. Moreover, with a nonlinear solution it is possible to solve the localization and classification problem with a minimum amount of information, which implies less communication load and a higher success probability. Based on its good performance in the simulations and its simplicity, the Simplex Downhill (SD) algorithm was implemented on the ESB sensor boards.

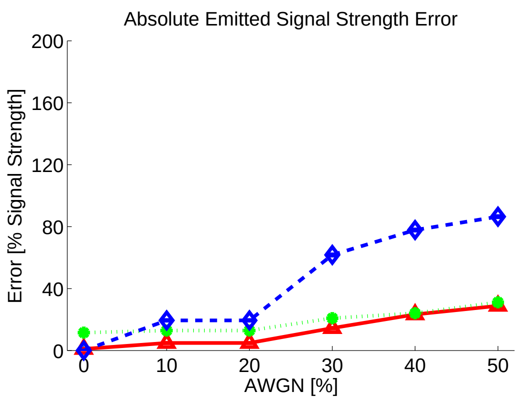
2) Localization Performance in Real-World Experiments: For the real-world experiments the same setup as for the simulations was used. The SD algorithm has been adapted from [16]. In contrast to the simulations, the event was not randomly placed in the event area, but at specific positions: P1(250, 250), P2(250,188), P3(188,188), and P4(219,219). The sensor node locations (**o**) and the event locations (**x**) are shown in Fig. 15.

Again two light sources of 25 Watt and 40 Watt have been used. Each location estimation has been performed 50 times. The localization was performed two times per second. The distance error means (μ) and the standard deviations (σ) of the localization tests are shown in table I.

Considering the distance of 125 cm between two neighbor nodes, a maximum mean location estimation error of 21 cm, at location P1 using the 40 Watt bulb, is acceptable. The SD method performs best for locations inside the square. The performance is decreased if the event position is very close



(a) Distance Error



(b) Signal Strength Error

Fig. 14: Accuracy of LLS, SD, and CG in an over-determined system.

TABLE I: Distance Error and standard deviation

Position	25 Watt		40 Watt	
	μ	σ	μ	σ
P1	18.43	0.14	20.91	0.23
P2	3.86	0.59	14.94	3.21
P3	6.3	0.85	4.13	0.11
P4	3.69	1.6	5.04	1.68

to a sensor node. The standard deviation in all experiments is very small.

Apart of the position, the SD method also computes the emitted signal strength of the event source. For the classification of events this value is even more important than the event position, as it is, assumed to be, characteristic for the event. The mean amplitude computed for the 25 Watt bulb is $1.71 \cdot 10^{-6}$ with a standard deviation of $0.246 \cdot 10^{-6}$. On the other hand, the mean amplitude of the 60 Watt bulb is $2.88 \cdot 10^{-6}$ with a standard deviation of $0.452 \cdot 10^{-6}$. Obviously, the resulting spectrums of both events are disjoint and can therefore be used for classification.

VII. Conclusions and Future Work

The DELTA algorithm provides an efficient and fast event detection and tracking algorithm as well as an accurate and distributed localization method. Tracking groups are created dynamically. DELTA works in many cases including smart dust environments with small radio ranges and high sensing ranges. The leader election procedure of DELTA is adaptive, quick and precise. Using the sensor readings improves both, the event detection and the tracking performance. The implementation of a moving average filter allows the suppression of bad located sensor nodes. Though, the convergence of the filter needs to be considered.

DELTA supports accurate in-network event localization. The evaluation has shown that a nonlinear algorithm is best suitable in terms of communication load and accuracy. The accuracy of the event localization might be improved using customized hardware. The TSL245 implemented on the ESB sensor boards is an infrared to frequency converter and, therefore, not best suited for visible light.

In future work we will make use of the computed event characteristics. In particular of the computed amplitudes. Based on training sets of different event sources at different event locations, classes of event amplitudes can be learned. Therefore, clustering mechanisms, e.g., a fuzzy k-means clustering algorithm, might be applied at the base station. In presence of events with multiple characteristics, e.g., sound and vibration, the cluster learning procedures could even be used to design advanced classifiers such as a fuzzy logic controllers (FLC). This FLC system could then be distributed to the sensor nodes enabling online in-network classification.

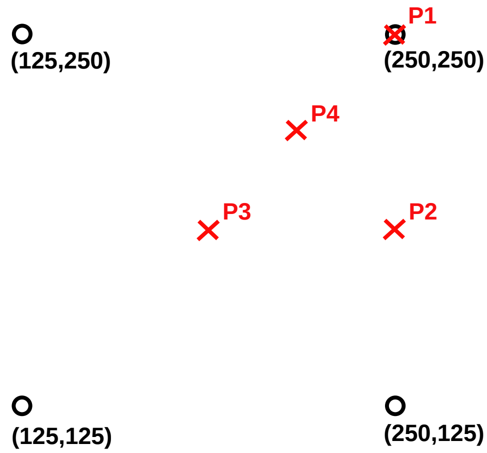


Fig. 15: Arrangement of sensors and event locations.

References

[1] T. Abdelzaher, B. Blum, D. Evans, J. George, S. George, L. Gu, T. He, C. Huang, P. Nagaraddi, S. Son, P. Sorokin, J. Stankovic, and A. Wood. Envirotrack: Towards an environmental computing paradigm for distributed sensor networks. In *Proc. of 24th International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, Mar. 2004.

[2] R. R. Brooks, P. Ramanathan, and A. M. Sayeed. Distributed target classification and tracking in sensor networks. *Proc. IEEE*, 91(8):1163–1171, August 2003.

[3] C.-C. G. Chang, W. E. Snyder, and C. Wang. Robust localization of multiple events in sensor networks. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, pages 168–177, 2006.

[4] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3):293–313, 2002.

[5] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem. Distributed online localization in sensor networks using a moving target. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 61–70, Berkeley, California, USA, 2004. ACM Press.

[6] S. Guha, R. N. Murty, and E. G. Sirer. Sextant: A unified node and event localization framework using non-convex constraints. In *MobiHoc'05*, pages 205–216, Urbana-Champaign, Illinois, USA, May 2005.

[7] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli. Optimized stateless broadcasting in wireless multi-hop networks. In *Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–12, Barcelona, April 23-29 2006.

[8] T. Instruments. Infrared light-to-frequency converter, 2008.

[9] M. Kumar, L. Schwiebert, and M. Brockmeyer. Efficient data aggregation middleware for wireless sensor networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 1579–1581, Fort Lauderdale, Florida, USA, October 25-27 2004.

[10] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitive comparison. *Computer Networks*, 43(4):499–518, 2003.

[11] D. Li and Y. H. Hu. Energy-based collaborative source localization using acoustic microsensor array. *EURASIP Journal on Applied Signal Processing*, 3:321–337, 2003.

[12] D. Li and Y. H. Hu. Least square solutions of energy based acoustic source localization problems. In *Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW'04)*, pages 443–446, Washington, DC, USA, 2004.

[13] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed. Detection, classification and tracking of targets. *IEEE Signal Processing Magazine*, 19(2):17–29, March 2002.

[14] L. Luo, T. F. Abdelzaher, T. He, and J. A. Stankovic. Envirosuite: An environmentally immersive programming framework for sensor networks. *ACM Transaction on Embedded Computing System (TECS)*, V:1–31, 2006.

[15] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[17] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 20–27, Berkeley, California, USA, 2004.

[18] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001.

[19] Scatterweb. Sensor platform, 2007. http://www.scatterweb.net.

[20] X. Sheng and Y.-H. Hu. Energy based acoustic source localization. In *The 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, 2003.

[21] X. Sheng and Y.-H. Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1):44– 53, January 2005.

[22] J. O. Smith and J. S. Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(12):1661– 1669, December 1987.

[23] A. Varga. Omnet++ simulator, 2006. http://www.omnetpp.org/.

[24] M. Waelchli, P. Skoczylas, M. Meer, and T. Braun. Distributed event localization and tracking with wireless sensors. In *5th International Conference on Wired/Wireless Internet Communications (WWIC '07)*, pages 247–258, Coimbra, Portugal, May 2007.

[25] M. Wälchli, M. Scheidegger, and T. Braun. 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 ménuires, France, January 2006.

[26] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(1):61–91, 2004.

Markus Wälchli received his Master Degree in Computer Science from University of Bern in January 2005. Since February 2005 Markus Wälchli has been employed as a PhD student at University of Bern. He participates in 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. His main research interests are event detection, tracking, and classification in wireless sensor networks. Furthermore, he is doing research in medium access control and topology control in sensor networks.

Samuel Bissig is currently doing his Master degree in Computer Science in the Computer Networks and Distributed Systems group at the University of Bern. In his master thesis he works on implementing an event detection, tracking, and classification framework on tiny sensor nodes. Further research interests are routing and medium access control in sensor networks.

Michael Meer received his Master Degree in Computer Science from University of Bern in November 2006. Since September 2006 Michael Meer has been working as a Software Engineer in Swisslog AG. His main research interests include efficient event tracking and routing in wireless sensor networks. Further interest include software engineering and reengineering.

Torsten Braun received the diploma and the PhD degrees from the University of Karlsruhe, Germany, in 1990 and 1993, respectively. From 1994 to 1995, he was a guest scientist with INRIA Sophia Antipolis. From 1995 to 1997, he worked as a project leader and senior consultant at the IBM European Networking Center, Heidelberg, Germany. Since 1998, he has been a full professor of computer science at the Institute of Computer Science and Applied Mathematics (University of Bern, Switzerland), heading the Computer Networks and Distributed Systems research group. Since 2007 he has been director of the Institute of Computer Science and Applied Mathematics. He has been a board member of SWITCH (Swiss Education and Research network) since 2000. During his sabbatical in 2004, he was a visiting scientist at INRIA Sophia-Antipolis and the Swedish Institute of Computer Science at Kista.