Distributed Data Fusion in a Ground Sensor Network

Mikael Brännström, Ron Lennartsson, Andris Lauberts,
Hans Habberstad, Erland Jungert, Martin Holmberg
Swedish Defence Research Agency (FOI)
P.O. Box 1165
SE-581 11 Linköping
Sweden
{mikbra, ron, andris, hanhab, jungert, mah}@foi.se

Abstract - *In order to fuse data from a distributed sensor system, different fusion architectures can be chosen. In this work, distributed data fusion has been applied to data from a ground sensor network. Measurements have been made using five seismic and five acoustic sensor nodes, each consisting of three sensors, on vehicles of five different types. The data were analyzed off-line to achieve tracking and classification of the vehicles. The tracking was made using standard Kalman filters, and classification was made by a combination of a Bayes' classifier and voting. The information system, which made distributed data fusion possible, was based on a combination of intelligent agents and a distributed look-up table for data storage and retrieval. The results indicate that this system can operate on-line with a high accuracy in both tracking and classification.*

Keywords: Unattended Ground Sensors, Distributed Data Fusion, Tracking, Classification.

1 Introduction

Autonomous sensor systems require an information system that can handle data without the need of an operator. For a system where all sensors are permanently connected, the choice of sensors and/or fusion algorithm can be decided beforehand since all possible combinations of sensors are known. In a system where sensors can be added or removed at any time, the information system (in each time instant) autonomously has to be able to establish which sensors that carry relevant information before performing data fusion. The information system also has to make the information available to the user by organizing the fused results in an efficient way. In recent years, there have been many publications on information processing in sensor networks, see e.g. [1-8] for an overview. References [1-3] point to whole volumes of conference proceedings or special issues. In short, papers [4-8] discuss energy saving methods, detection and tracking of targets, data fusion and classification. In [4] one leading sensor (the agent) handles energy conservation by minimizing communication of information. Energy saving by switching sensor nodes on-off and by randomization of data transfer is discussed in [5]. To evaluate the coverage of the region by a sensor network, energy exposure is used to detect intruding targets, see [6]. The sometimes subtle difference between a decentralised network (fusion locally at each node) and a distributed network (information "hopped" through to some central fusion site) is pointed out in [7]. Here we also find decision and fusion algorithms cast in information form, further discussed in our section 2.2.2. Classification, finally, is discussed in [8], where it is found that a suboptimal decision fusion classifier could be an attractive choice even for data with some dependency.

In this work, we have developed an information system for data from an unattended ground sensor (UGS) network. We have thus concentrated our efforts to solve some of the problems that arise in this particular application since a complete solution to all problems related to distributed data fusion would simply be too big of a task. Particular attention has been paid to the choice of sensors.

The goal is to detect, classify, and track vehicles moving within the area covered by the UGS network. Each single sensor can detect the sound (acoustic sensor) or vibrations (seismic sensor) caused by vehicles in their vicinity. Specific signatures (a compressed representation of the raw data) for each of the vehicles can be calculated from the sensor signal. The signature can then be used to make a classification of the vehicle (single-node classifier). Furthermore, by placing three identical sensors in one sensor node, it is possible to calculate also the direction to the sound source. All these calculations can be made in one single sensor node, and do not require a distributed information system.

In order to utilize data from several sensor nodes, the information system needs to determine which sensor nodes are measuring on the same target (association). By combining data from different sensor nodes, the position of the vehicle can be found by triangulation using the directions calculated in different sensor nodes. New tracks should be initiated when and only when no other node is tracking that vehicle. For classification, the information system can improve the performance of the system compared to the single-node classifiers by performing statistical analysis of the single-node results. It is also possible, and sometimes advantageous, to use the signatures calculated in each node and make a model based on data from several nodes to classify the vehicle (multi-node classifier).

2 Distributed fusion architecture

In an unattended ground sensor network for ground surveillance there are a number of issues that need to be handled:

- Automatic data association (i.e. decide which sensors that are measuring on the same target)
- Sensor selection (i.e. exclude the sensors with poor signal quality)
- Data fusion (i.e. tracking and classification)
- Information lookup
- Robustness and scalability

In order to obtain robustness and scalability the network must handle sensor nodes to be added and removed at any time from the network. Removing a single sensor node should not make the sensor network as whole stop working, i.e. centralized solutions without redundancy should be avoided. Adding more sensor nodes should not degrade the performance of the sensor network, e.g. by overloading the network or a single sensor node. Regarding these facts and that the sensor network itself is physically distributed, a distributed solution is preferred.

Tracking, classification and sensor selection is made by mobile agents, called track agents, which move through the network as the tracked target moves. Signal processing, such as bearing calculations, feature extraction, single sensor classification is done locally in each sensor node and the compact result is sent to the track agent that fuses the information. The track agent sends the fused track data (position, velocity, type, etc.) to the interested user(s), as shown in Fig. 1.

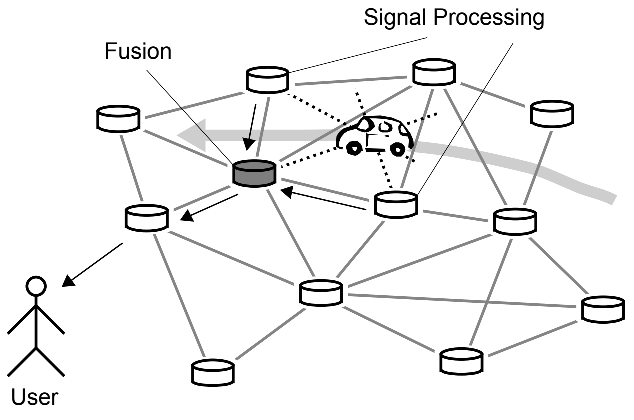


Fig. 1. Dataflow in the sensor network. Track agent in grey sensor node.

Data association in a distributed sensor network requires that some kind of information lookup service exists since all information is not directly available in any single node (due to scalability issues). Thus, it must be possible for a sensor to lookup possible matching tracks. The information lookup service is also needed for the users to find tracks in the network (or the other way around – track agents finding interested users).

2.1 Sensor signal processing

In this section the signal processing that is carried out locally in each sensor node, is described. Once a vehicle is detected by a sensor node, two calculations are made in order to determine in which direction the source is located and to classify what type of source it is. Both of these calculations require the use of time series, and for that purpose the data are divided into observation windows of length 0.5 seconds. All calculations in the sensor node are then made using one observation window at the time.

2.1.1 Estimation of direction of arrival

The direction to a source is estimated with a circular array with three sensors, see Fig. 2. The direction to a source, θ, is estimated by studying the cross correlations between the signals measured by the three sensors in each node, together with the assumption of planar waves. The cross correlation between the signal in sensor i and j is given by

$$C_{ij}(k_{ij}) = \frac{1}{N}\sum_{n=1}^{N} x_i(n)x_j(n-k_{ij}), \qquad (1)$$

where N is the window length and $x_i(n)$ is the raw data measured by sensor i. The number of time steps k_{ij} that results in the maximum cross-correlation corresponds to the difference in time of arrival for that sensor pair. The sensor pair with the smallest difference in time of arrival is then used to calculate the direction to the source. For example, if the smallest difference in time of arrival is found between signals in sensor 2 and 3 the direction to the source θ is given by

$$\theta = \arcsin\left(\frac{ck_{23}T}{2r\cos(\pi/6)}\right), \qquad (2)$$

where c is the propagation speed and T is the sample interval used.

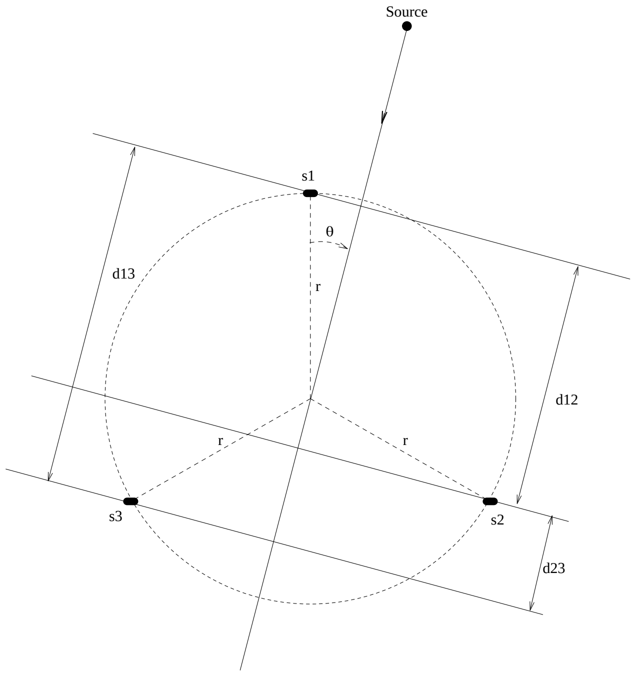


Fig. 2. A circular array with three sensor $s1$, $s2$ and $s3$ separated by 120 degrees and with radius r. The direction to the source θ is estimated and later used for triangulation in the network.

2.1.2 *Classification*

The classification process can be viewed as a two-step process, feature extraction and partitioning of the feature space.

First, features are extracted from the acoustic and seismic signals. The features have to represent the signal in a compact way and most importantly maintain enough information to perform classification of the vehicles. In this work an auto-regressive (AR) model is used to extract features. An AR-model of order p can be written as:

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + e(n)\,, \tag{3}$$

where $x(n)$ is the measured signal and $e(n)$ is white noise with variance σ^2. The signal is divided into observation windows of length N, and the model is fitted to each observation window by Yule-Walker's method [9]. The model parameters a_1 to a_p are used as features for classification purposes.

In order to perform the actual classification we have chosen a Bayesian approach to build a minimum-error-rate classifier. There are many ways to represent such a classifier, one way is by a set of discriminant functions $g_i(\mathbf{A})$ [10]. Where $g_i(\mathbf{A}) = P(\omega_i|\mathbf{A})$ so that the maximum discriminant function corresponds to the maximum a posterior probability. Assume that we have s signal classes $\omega_i,\ldots,\omega_s$. Classification can then be made by assign a feature vector $\mathbf{A} = (a_1,\ldots,a_p)^t$ to class ω_i if

$$g_i(\mathbf{A}) > g_j(\mathbf{A}) \quad \text{for all} \quad j \neq i\,. \tag{4}$$

Further, if we assume that the feature distributions are multivariate normal and that the a priori probabilities of all classes are equal. Then the discriminant function for class ω_i is given by

$$g_i(\mathbf{A}) = -\frac{1}{2}(\mathbf{A}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{A}-\boldsymbol{\mu}_i) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i|\,. \tag{5}$$

Where $\boldsymbol{\mu}_i$ is the mean and $\boldsymbol{\Sigma}_i$ is the covariance matrix for class ω_i. In practical cases the means and the covariance matrix are unknown and have to be estimated from a training set.

2.2 Multi sensor fusion

By data fusion is meant a process that combines data and knowledge from different sources with the aim to maximize the value of data, possibly uncertain, incomplete or contradictory. Frequently the sensors work in complementary ways with raw data in different formats - in our case sensed by microphones and geophones. In these cases data fusion is preferably carried out at a higher level of information content. Alternatively, extracted features, observed by each sensor, are fused into a common unit. Data fusion involves several cooperating parts, partly overlapping. Detection and classification may be done by independent sensors at different times. Association makes an important point for a successful fusion of sensor data. The common information is here fused to form a clearer picture of target position and identity.

2.2.1 *Mobile Agents*

Tracking and classification is done by mobile agents, called track agents. The particular algorithms for tracking and classification are described in the following sections. Here the process of creating, destroying and moving the agents as well as the association mechanism is described.

There exists one track agent for each target being tracked by the sensor network. Thus, when a new target is detected by a sensor a new track agent is created. The track agent will from that point on track the corresponding target until the target moves out of the coverage area of the sensor network, at which point the track agent will be removed and the track data will be stored for historical reasons only. The track agent moves from node to node in the network as the target being tracked moves. The movement scheme is quite simple; the agent moves to the node closest to the target being tracked. This way the communication distances between the track agent and other sensor nodes observing the target is minimized. This is also an important feature which makes it easier to find the agent in the network (e.g. by using geographic routing).

The association burden lies on the sensor node detecting a target. Whenever a sensor detects a target it does a local track agent lookup (described further down) to find the track agents in the local area. If a matching track agent is found the sensor data is sent to that agent, otherwise a new track agent is created to which the sensor data is sent. Each sensor node tries to do a sensor data to sensor data association in order to minimize the track agent lookup process. Instead of associating single target observations to a track, series of target observations are being associated to a track.

The track agent automatically receives sensor data from the sensor nodes that observe the target being tracked. In this process it also gets notified of which sensor nodes that are within range of the target. In order to control the data flow of sensor data being received the track agent can suppress sensor nodes which are not needed and control the data rate of the other sensor nodes. This is of course a tradeoff between track quality and resources such as network bandwidth and power consumption.

2.2.2 *Tracking*

This study incorporates data fusion in tracking by computing positional information using widely distributed sensors. A decentralized data fusion system, such as the one proposed, is spanned by a network of sensor nodes, each possessing its own processing power. In such a system data fusion is carried out locally at each node based on own observations and conveyed information from nearby nodes. No single sensor node has global knowledge of either the network topology or ways of communication. A system built in this way may be freely scaled to other sizes, is not vulnerable to smaller losses of nodes (redundancy), and may easily be upgraded by modular units. In the following we give the basic equations for a Kalman filter in its decentralized form, see [11] combined with [12] and [13]. Let

$\mathbf{x}(k)$	State vector with time index k
$\mathbf{Z}^k = \{\mathbf{z}(1),...,\mathbf{z}(k)\}$	A sequence of observations
$\hat{\mathbf{x}}(i\|j) = \mathrm{E}\{\mathbf{x}(i)\|\mathbf{Z}^j\}$	Estimate of state vector
$\tilde{\mathbf{x}}(i\|j) = \mathbf{x}(i) - \hat{\mathbf{x}}(i\|j)$	Difference to state estimate
$\mathbf{P}(i\|j) = \mathrm{E}\{\tilde{\mathbf{x}}(i\|j)\tilde{\mathbf{x}}^T(i\|j)\|\mathbf{Z}^j\}$	Covariance of state estimate

The information form of the Kalman filter is derived by rewriting the state vector estimate and covariance in two new variables

$$\hat{\mathbf{y}}(i|j) \equiv \mathbf{P}^{-1}(i|j)\hat{\mathbf{x}}(i|j), \tag{6}$$

$$\mathbf{Y}(i|j) \equiv \mathbf{P}^{-1}(i|j). \tag{7}$$

Multiple sensor observations are assumed to follow the model

$$\begin{aligned} &\mathbf{z}_i(k) = \mathbf{H}_i(k)\mathbf{x}(k) + \mathbf{v}_i(k), \quad i = 1,...,N \text{ sensors}, \\ &\mathrm{E}\{\mathbf{v}(i)\mathbf{v}^T(j)\} = \delta_{ij}\mathbf{R}(i), \end{aligned} \tag{8}$$

with matrices

$\mathbf{H}_i(k)$ transform state vector to observation,

$\mathbf{v}_i(k)$ measurement noise,

$\mathbf{R}(k)$ covariance.

The information associated with an observation can then be written in the form

$$\mathbf{i}_i(k) \equiv \mathbf{H}_i^T(k)\mathbf{R}^{-1}(k)\mathbf{z}_i(k), \tag{9}$$

$$\mathbf{I}_i(k) \equiv \mathbf{H}_i^T(k)\mathbf{R}^{-1}(k)\mathbf{H}_i(k). \tag{10}$$

With these definitions the estimate and covariance are updated as

$$\hat{\mathbf{y}}(k|k) = \hat{\mathbf{y}}(k|k-1) + \sum_{i=1}^{N}\mathbf{i}_i(k), \tag{11}$$

$$\mathbf{Y}(k|k) = \mathbf{Y}(k|k-1) + \sum_{i=1}^{N}\mathbf{I}_i(k). \tag{12}$$

Evidently this update is done in terms of dimension of the state vectors, in contrast to a standard Kalman filter which is updated in terms of the dimension of the observations. The advantage of the information filter is evident in multisensor problems where the update is carried out through a sum of uncorrelated terms $\mathbf{i}_i$ and $\mathbf{I}_i$, which is not possible using the Kalman filter in its standard form. It is the *tracking agent's* responsibility to keep the information transfer (i.e. summations over $\mathbf{i}_i$ and $\mathbf{I}_i$) at a minimum but still confidently track the target.

The prediction of information is done according to (sensor index i suppressed to simplify notation)

$$\hat{\mathbf{y}}(k|k-1) = \mathbf{L}(k|k-1)\hat{\mathbf{y}}(k-1|k-1), \tag{13}$$

$$\mathbf{Y}(k|k-1) = \begin{bmatrix} \mathbf{F}(k)\mathbf{Y}^{-1}(k-1|k-1)\mathbf{F}^T(k) \\ +\mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}^T(k) \end{bmatrix}^{-1}, \tag{14}$$

where

$$\mathbf{L}(k|k-1) = \mathbf{Y}(k|k-1)\mathbf{F}(k)\mathbf{Y}^{-1}(k-1|k-1), \tag{15}$$

with matrices

$\mathbf{F}(k)$ transfer for state vector,

$\mathbf{Q}(k)$ process noise,

$\mathbf{G}(k)$ transfer for process noise.

Track initiation is prepared by choosing "rule-of-thumb" parameter values for measurement noise, process noise, state information, state estimate and covariance. In particular, the state information starts at zero, state estimate is flagged as undecided and its covariance set to a diagonal matrix with small elements. Typically, a new target is first observed by a single sensor situated at the outskirt of the network. The observation uncertainty ellipse, defined by the co-rotated $\mathbf{R}$ matrix, outlines an elongated region in the bearing direction. The first position estimate lies at the centre of this ellipse. Subsequent observations from the same sensor add more information, usually with the effect to artificially "attract" the target to the sensor. Eventually, other sensors contribute independent bearings that greatly improve the tracking. An important point is the association of observation to target. Here, valid observations are gated within a maximum window defined by a normalised distance, $(\mathbf{z}-\hat{\mathbf{x}})^T\mathbf{P}_p^{-1}(\mathbf{z}-\hat{\mathbf{x}})$ < *maxgate*, where $\mathbf{P}_p$ is a submatrix of the state covariance, containing only the position covariance. However, complications arise for the case of close targets. Given the probability that a particular target is observed (e.g. by feature classification) it is possible to weigh individual terms in the summation of $\mathbf{i}_i$ and $\mathbf{I}_i$, and thereby improve the track-to-target associations.

2.2.3 Classification

After association has been made correctly, there are several ways of performing classification using multi-sensor data fusion. We have chosen two different approaches:

- All single-node classifications during the passage (i.e. from all sensors and from all observation windows that have enough signal energy) can be used in a voting scheme. This means that correct classification can be made even if the model does not always make the correct decision, as long as the majority of classifications made are correct.
- A multisensor classifier can be used, based on the model parameters of the AR-model described above. This can then be combined with a statistical

analysis of all classifications made during one passage.

At this point, only the voting scheme has been implemented, but also a multisensor classifier will be implemented in the near future.

2.3 Distributed lookup

An unattended ground sensor network requires a lookup service for at least two reasons:

- Users need to find information (tracks) in the network (global lookup)
- Sensors detecting something need to know what targets are being tracked in its surrounding (local lookup)

The traditional way of solving this problem is to use a central server, either determined a priori or chosen by election, where information (or reference to) is registered and looked up. Search engines on the Internet as well as the Jini technology [14] in its basic form are examples of this approach. This solution is however not scalable enough, at least not in the sensor network context. When the network grows, the centralized server will at some point get overloaded. Since the sensor nodes run on battery power the server node will also run out of power much faster than the average node in the network. At some level the server node's bandwidth will get consumed by the registration and lookup messages and replies sent to and from the server. Thus, the central server node would become a bottleneck of the entire sensor network, and if this node fails (e.g. by power depletion, hardware failure or destroyed) the sensor network would stop working. In the following subsections distributed methods for solving the global and the local lookups are described.

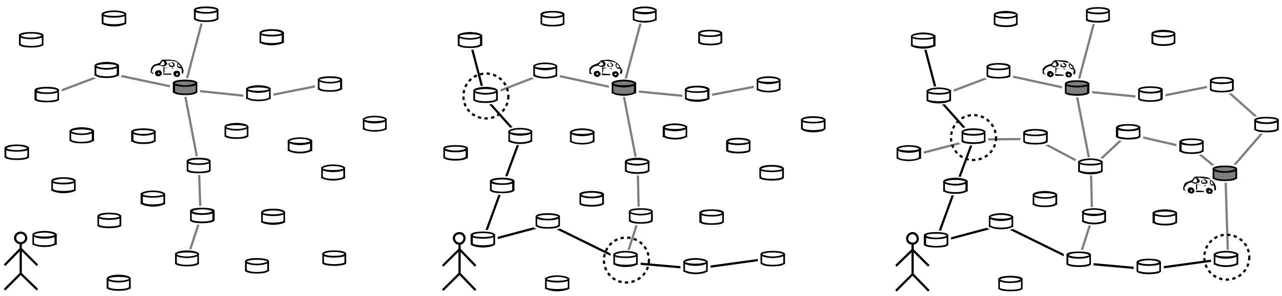
2.3.1 Global lookup

The global lookup is designed like a match-making facility. Users want to find tracks (or track agents) in the network, and track agents want to find users that are interested in the information the track agent holds. Thus, the registration process and the lookup process are combined into a single process. On the user side this can be seen as a subscription (e.g. the user wants updates of all track movements every minute). The user will do a lookup of all tracks and register its interests in case more tracks are detected in the future. The track agents will register themselves upon creation and do a user lookup of the registered user subscriptions.

The match-making is based on so called pseudo quorums [15]. A pseudo quorum is a subset of a collection of objects, where *every two* pseudo quorums intersect with *high probability* (compared to regular quorums which are guaranteed to intersect). The idea is that a node that wants to register some information, sends a registration message in a couple of directions. When another node wants to lookup this information it sends a lookup message in a couple of directions from that node. The lookup message will then with high probability reach a node which has received the registration message [16]. Fig. 3 illustrates how a user finds and gets notified about all tracks in the network using pseudo quorums.

When a track agent in some node in the network determines to register itself, it sends registration messages in a number of directions (a design parameter) containing information about the track, the node address. Geographic routing techniques [17] are used to send the message in straight lines in order to increase the probability of intersections. This is shown in Fig. 3a. When a user connects to a node in the network it will send a subscription message in a number of directions. This message contains the node address of the user and some subscription parameters (e.g. vehicle types of interest, area of interest, update frequency). When the subscription message arrives at a node which has previously received a matching track registration (see the circles in Fig. 3b), a message is sent to the user. The user can then communicate directly with the track agent for more information. When another vehicle is detected by the sensor network a new track registration is made, in which the user subscription will be found (see Fig. 3c). This will also result in a message being sent to the user as in the previous case.

The registrations are time limited so that deregistration is not necessary. Instead registrations need to be renewed regularly and also when a track agent moves from one node to another. An optimization that can be made is to leave a trail in the node the track agent is leaving from instead of registering the track in the entire network. This way messages sent to the track agent's old location can easily be routed to the new location. This optimization also applies to users moving in the network and reconnecting to another node.



(a) (b) (c)

Fig. 3. Global Lookup. (a) Track registration, track agent in grey sensor node. (b) User lookup of tracks, circles denotes matches. (c) Another track agent registers itself and finds the user.

2.3.2 *Local lookup*

The local lookup is needed when a sensor observes something and needs to associate this with the corresponding track somewhere in the network, if any. The geographic information in the observation is used to find the track agents that are possible matches. In the case of microphone arrays and geophone arrays, this means the bearing and in practice some finite distance. Since the track agent resides in a sensor node close to the target being tracked, an optimized version of the global lookup can be used. This means that a lookup message is sent in the direction of the bearing and a reply is sent back to the sensor for each possible matching track agent.

3 Measurements

A field trial based on passages of military vehicles through a network was carried out to obtain real sensor data for network processing. The network was placed near a dirt road in a forest-like environment with some open areas. The selected area includes the problems with direct, reflecting and damping of sound propagation that must be handled in an unattended ground sensor network. The total span of the network was near 400 m in length and 200 m in width with five acoustical sensor nodes, five seismic sensor nodes and two laser sensors. The laser sensors were used for reference positioning of vehicles. The design of the network system is based on a 32 channel National Instruments PXI-1000B and PXI4472 data acquisition system. Differential audio cables with a length of 200 m where used to connect to the microphone/geophone amplifiers close to the acoustical and seismic nodes.

All sensors were amplified at the node by a microphone/geophone amplifier with amplification up to 70 dB. At the output of the amplifier a differential line driver was placed for noise suppression in the system. A sampling rate of 48 kHz with 16 bit resolution, and a frequency interval from 60 Hz to 22 kHz was used to achieve as high quality raw data as possible.

Measurements were performed with five nodes of microphones and five nodes of geophones, each node carrying three sensors of the same kind at the corners of equilateral triangular arrays. The radii of circles circumscribing the sensor arrays are based on the desired angle resolution, sampling rate and propagation speed. In this network configuration we aimed for small easy to handle acoustic sensors with an angle resolution of better than three degrees. The acoustic sensor node has the shape of a discus which is based on two aluminum halves mounted together with three electret microphones mounted inside. Each microphone is mounted in a mill cut rail spaced by 120 degrees angle from each other and at a distance of 100 mm from the center of the discus.

The propagation of sound is most depending on the air temperature which varied between 3.9 to 12.1 degrees Celsius at the field trials. The temperature variation changed the speed of sound from 334 m/s to a maximum at 339 m/s which was accounted for and used in the calculations of angles and positions.

The configurations of the geophone nodes are similar to the acoustical nodes with three sensors in a triangular formation. The propagation of vibrations in the ground varied from 150 m/s to a maximum of 3000 m/s and was measured by generating impulses from a given incident angle and the velocity could be calculated by the time difference between the sensors. The seismic sensors were attached to ground of varying condition, such as bedrock, requiring drilling, or a loose marsh like ground.

Each vehicle passed, one at the time, through the network in five different runs; twice at low speed (going east and west), twice at high speed (east and west), and once at variable speed. [18]

4 Results

The sensor network was used to record signals during all passages. These signals were then used in the information system to calculate tracks and classifications. The calculations were made off-line, but the algorithms are fast enough to be used on-line. To simulate the effect of a wireless communication network, a simulator was developed to provide estimations of transmission time delays.

Figs. 4 and 5 show estimated tracks of two vehicles sensed by five microphone nodes and one geophone node (four additional geophone nodes malfunctioned and were dismissed). The total path is about 450 meters. Computations, using Kalman filter in information form, are based on bearings only. Deviations from the true path (dashed line indicating road) are mostly due to 1) bias in node positions and orientations, 2) deviating bearings caused by natural background disturbances, and 3) some not accounted time delays in received signals. The sudden maneuvers of the path in Figures 4 and 5 is due to one or more of the reasons mentioned above.

The RMS position errors, based on the mean shortest distance to the road, turn out as 10 and 13 meters for the wheeled vehicle and T72 tank, respectively. Velocity errors are defined with respect to a mean speed, calculated by timing the traverse at the two laser triggers, L1 and L2, yielding 1.3 m/s for the wheeled vehicle (the tank varied its speed too much to yield a sensible error). Looking at the cases shown in these figures (and most other cases as well) we see a general trend of estimated tracks lying below the road at start, climbing up in the middle part, then lying below the road at the end again. This common pattern could be an indication of a bias in node positions, and thereby unnecessarily increasing the position error.

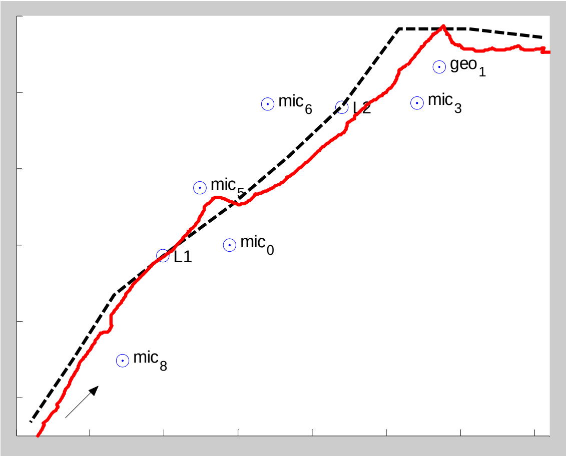


Fig. 4. Light wheeled vehicle running in the direction of the arrow at speed 23 km/h (solid line). The road is shown as a dashed line. Microphone (mic) and geophone (geo) nodes are marked with circles.

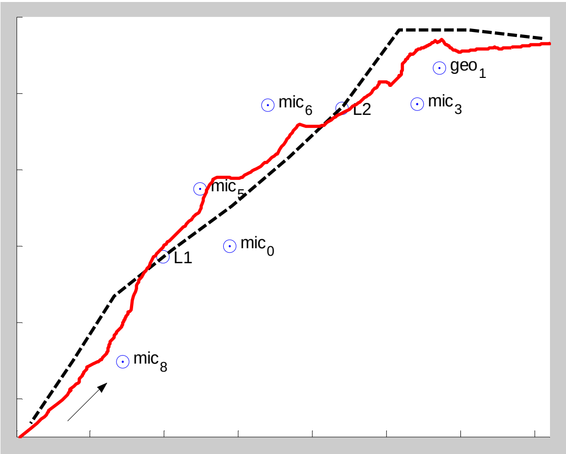


Fig. 5. T72 tank running at mean speed 37 km/h.

In order to evaluate the classification ability, segments containing the maximum power were automatically selected from each recording. The length of the segments was also automatically selected such that the power decays to five percent of the maximum at the beginning and end of each segment. The data segments were then divided into 0.5-second observation windows with no overlap. Next the features were extracted with an AR-model with $p = 27$ (a greater p does not improve the classification performance while a smaller p reduces the classification performance) and finally a six-class minimum-error-rate classifier was implemented. To train the classifier a randomly selected subset (70%) of the features were used. The remaining 30% were then used to evaluate the classifier. The results are summarized in the confusion matrix displayed in Table 1. As can be seen the method provides a high level of correct classification for all six classes.

Table 1: Class 1 and 2 are different light wheeled vehicles, class 3 is a heavy wheeled vehicle, classes 4 and 5 are different light tracked vehicles, and class 6 is a heavy tracked vehicle

Output→ Input ↓	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
Class 1	100%	0%	0%	0%	0%	0%
Class 2	0%	100%	0%	0%	0%	0%
Class 3	1%	0%	99%	0%	0%	0%
Class 4	0%	0%	0%	99%	1%	0%
Class 5	1%	0%	1%	3%	95%	0%
Class 6	0%	0%	0%	0%	0%	100%

In a real situation, the vehicle may be classified twice per second during the whole passage through the sensor network. This means that even though some misclassifications are made during a passage, the overwhelming majority will be correct, and thus perfect classification can be achieved in this case.

5 Conclusions

Measurements have been made on passages of six different types of vehicles using a ground sensor network. An information system utilizing distributed data fusion was developed in order to track and classify the vehicles. The information system was based on intelligent agents and a distributed look-up table. This approach was successful in handling the data, and gave nearly perfect classification and tracking results. In this work, only passages with one vehicle at the time was used, which of course made association rather easy. However, it is our belief that also situations with several vehicles can be handled provided that the sensors have the capability to resolve the vehicles.

In the future, the data fusion methods will be refined. Particularly, the multi-sensor classification will be extended with true multi-sensor classifiers, not only voting among the sensors as is the case at the moment. Also, larger networks will be simulated to validate the information system at larger scales. Moreover, the information system will be coupled to a query language [19] to provide the user with an easy-to-use interface.

Acknowledgements

The project has been financed by the Swedish Armed Forces.

References

[1] Feng Zhao and Leonidas Guibas, editors. *Proc. Second Int. Workshop on Information Processing in Sensor Networks (IPSN 2003)*, Palo Alto, California, 22-23 April, 2003.

[2] Special Issue on Sensor Networks and Applications, *Proceedings of the IEEE*, 91(8), August 2003.

[3] Special Issue on Collaborative Signal and Information Processing in Microsensor Networks, *IEEE Signal Processing Magazine*, March 2002.

[4] C.-Y. Chong, F. Zhao, S. Mori, and S. Kumar. Distributed Tracking in Wireless Ad Hoc Sensor Networks. "Proceedings of the sixth international conference on information fusion", Cairns, Australia, pp. 431-438, July 8-11, 2003.

[5] S. Appadwedula, D. L. Jones, and V. V. Veeravalli. Energy-Efficient Detection in Sensor Networks. "Proceedings of the sixth international conference on information fusion", Cairns, Australia, pp. 764-771, July 8-11, 2003.

[6] T. Clouqueur, P. Ramanathan, and K. K. Saluja. Exposure of Variable Speed Targets Through a Sensor Field. "Proceedings of the sixth international conference on information fusion", Cairns, Australia, pp. 599-605, July 8-11, 2003.

[7] H. Durrant-Whyte and Ben Grocholsky. Management and Control in Decentralised Networks. "Proceedings of the sixth international conference on information fusion", Cairns, Australia, pp. 560-566, July 8-11, 2003.

[8] A. M. D'Costa and A. M. Sayeed. Data Versus Decision Fusion for Classification in Sensor Networks. "Proceedings of the sixth international conference on information fusion", pp. 889-894, July 8-11, 2003.

[9] J. G. Proakis and D. G. Manolakis. Digital Signal Processing. *Prentice and Hall*, 1996.

[10] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. *Johan Wiley & Sons*, 1973.

[11] E. Nettleton and H. Durrant-Whyte. Delayed and Asequent Data in Decentralised Sensing Networks. *Sensor Fusion and Decentralized Control in Robotic Systems IV, vol. 4571,* Gerard T. McKee and Paul S. Schenker, eds., SPIE Bellingham, Washington, 2001. http://www.acfr.usyd.edu.au.

[12] H. Durrant-Whyte and M. Stevens. Data Fusion in Decentralised Sensing Networks. *4th International Conference on Information Fusion,* Montreal, Canada, 2001.

[13] M. Ridley, E. Nettleton, S. Sukkarieh and H. Durrant-Whyte. Tracking in Decentralised Air-Ground Sensing Networks. *5th International Conference on Information Fusion*, Annapolis, U.S.A., 8-11 July, 2002.

[14] Jini. http://www.jini.org.

[15] Ilknur Aydin and Chien-Chung Shen. Facilitation match-making service in ad hoc and sensor networks using pseudo quorum. *11th IEEE Int. Conf. on Computer Communications and Networks (ICCCN)*, Miami, Florida, 14-16 October, 2002.

[16] David Braginsky and Deborah Estrin. Rumor routing algorithm for sensor networks. *1st ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, USA, September 28, 2002.

[17] Brad Karp and H. T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks. *Sixth Int. Conf. on Mobile Computing and Networking (MobiCom)*, Boston, MA, USA, 6-11 August, 2000.

[18] Hans Habberstad. Field trials with acoustical and seismological sensors in a network. FOI-R--1087--SE, Swedish Defence Research Agency (FOI), Sweden, 2003.

[19] S.-K Chang, G. Costagliola, E. Jungert, and F. Orciuoli. Querying distributed multimedia databases and data sources for sensor data fusion, *Journal of IEEE transaction on Multimedia,* accepted for publication.