Detection, Classification and Tracking of Targets in Distributed Sensor Networks

Dan Li, Kerry D. Wong, Yu H. Hu and Akbar M. Sayeed
Department of Electrical and Computer Engineering
University of Wisconsin-Madison, USA
dli, kerryw@cae.wisc.edu
hu, akbar@engr.wisc.edu

Abstract – *We outline a framework for collaborative signal processing in distributed sensor networks. The ideas are presented in the context of tracking multiple moving objects in a sensor field. The key steps involved in the tracking procedure include event detection, target classification, and estimation and prediction of target location. Algorithms for various tasks are discussed with an emphasis on classification. Results based on experiments with real data are reported which provide useful insights into the essential nature of the problems. Issues, challenges and directions for future research are identified.*

Keywords: collaborative signal processing, sensor networks, detection, target localization, classification, tracking

1 Introduction

Networks of small, densely distributed wireless sensor nodes are being envisioned and developed for a variety of applications involving monitoring and manipulation of the physical world in a tetherless fashion [1], [16], [17], [22], [23]. Typically, each individual node can sense in multiple modalities but has limited communication and computation capabilities. Many challenges must be overcome before the concept of sensor networks becomes a reality. In particular, there are two critical problems underlying successful operation of sensor networks: efficient methods for exchanging information between the nodes, and *collaborative signal processing (CSP)* between the nodes to gather useful information about the physical world.

This paper describes the key ideas behind the CSP algorithms for distributed sensor networks being developed at the University of Wisconsin (UW). We also describe the basic ideas on how the CSP algorithms interface with the networking/routing algorithms being developed at Wisconsin (UW-API) [2]. We motivate the framework via the problem of detecting and tracking a single maneuvering target. This example illustrates the essential ideas behind the integration between UW-API and UW-CSP algorithms and also highlights the key aspects of detection and localization algorithms. We then build on these ideas to present our approach to tracking multiple targets that necessarily requires classification techniques.

Tracking multiple targets via a wireless sensor network is a very challenging, multi-faceted problem and several research groups have tackled various aspects of it [3]-[8],[12],[13],[15],[18],[19],[21],[23],[25]. We consider the signal processing aspects of this problem under the constraints imposed by limited capabilities of the nodes as well as those associated with networking and routing. Consequently, in the present form, all our algorithms are based on processing a single sensing modality, such as seismic or acoustic. Furthermore, current detection and classification algorithms are based on single-node processing, whereas localization and tracking algorithms require collaboration between nodes. Our main emphasis in this paper is on *target classification* that is arguably the most challenging signal processing task in the context of sensor networks. We provide some numerical results based on real data that lend useful insights into the problem and help identify key issues and challenges. Finally, based on our findings we identify some promising directions for future research.

In the following two subsections, we motivate the need for CSP in sensor networks and describe the notion of space-time cells for facilitating CSP. Section 2 discusses the overall framework for detecting and tracking a single target. Section 3 describes classification algorithms that we have investigated and reports our experiences with real data. Section 4 identifies some issues and challenges and Section 5 discusses some directions for future research. Section 6 provides some concluding remarks.

1.1 Collaborative Signal Processing

Power consumption is a critical consideration in a wireless sensor network. The limited amount of energy stored at each node must support multiple functions, including sensor operations, on-board signal processing, and communication with neighboring nodes. Thus, one must consider power-efficient sensing modalities, low sampling rates, low-power signal processing algorithms, and efficient communication protocols to exchange information among nodes. To facilitate monitoring of a sensor field, including detection, classification, identification, and tracking of targets, global information in both space and time must be collected and analyzed over a specified space-time region. However, individual nodes only provide spatially local information. Furthermore, due to power limitation, temporal processing is feasible only over limited time periods. This necessitates CSP --- collaboration between nodes to process the space-time signal. A CSP algorithm can benefit from the following desirable features:

Distributive processing – Raw signals are sampled and processed at individual nodes but are not directly communicated over the wireless channel. Instead, each node extracts relevant summary statistics from the raw signal, which are typically of smaller size. The summary statistics are stored locally in individual nodes and may be transmitted to other nodes upon request.

Goal-oriented, on-demand processing – To conserve energy, each node only performs signal processing tasks that are relevant to the current query. In the absence of a query, each node retreats into a standby mode to minimize energy consumption. Similarly, a sensor node does not automatically publish extracted information --- it forwards such information only when needed.

Information fusion – To infer global information over a certain space-time region from local observations, CSP must facilitate efficient, hierarchical information fusion --- progressively lower bandwidth information must be shared between nodes over progressively large regions. For example, (high bandwidth) time series data may be exchanged between neighboring nodes for classification purposes. However, lower bandwidth CPA (closest point of approach) data may be exchanged between more distant nodes for tracking purposes.

Multi-resolution processing – Depending on the nature of the query, some CSP tasks may require higher spatial resolution involving a finer sampling of sensor nodes, or higher temporal resolution involving higher sampling rates. For example, reliable detection may be achievable with a relatively coarse space-time resolution, whereas classification typically requires processing at a higher resolution. Multiresolution space-time processing using wavelets [24] may be fruitfully exploited in this context.

1.2 Space-Time Sampling and Space-Time Cells

Each object in a geographical region generates a time-varying space-time signature field that may be sensed in different modalities, such as acoustic, seismic or thermal. The nodes sample the signature field spatially and the density of nodes should be commensurate with the rate of spatial variation in the field. Similarly, the time series from each sensor should be sampled at a rate commensurate with the required bandwidth. Thus, the rate of change of the space-time signature field and the nature of the query determines the required space-time sampling rate.

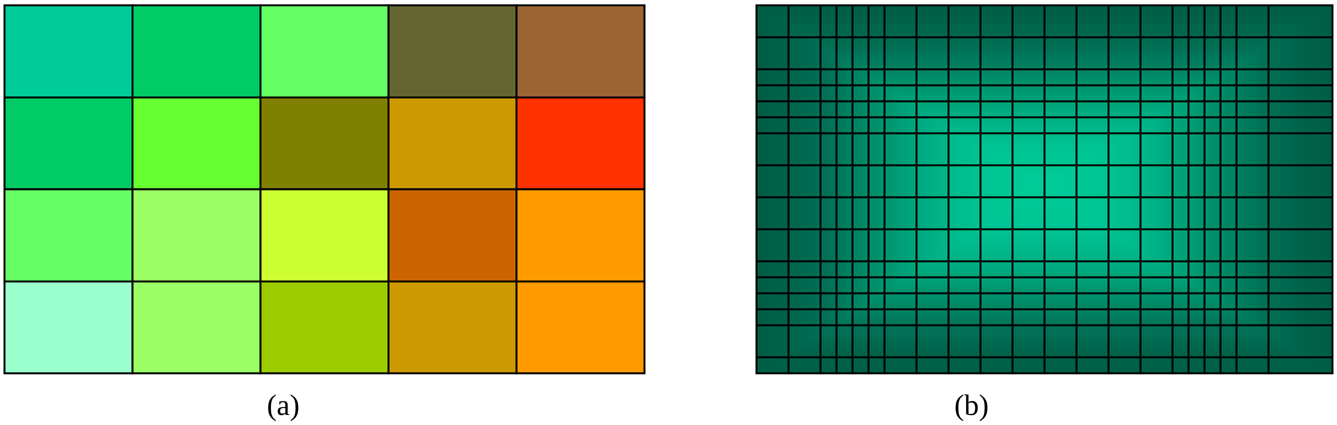

Figure 1: *A schematic illustrating the notion of space-time cells. Vertical dimension depicts space and horizontal dimension depicts time. Different shades depict the variations in the space-time signature field. (a)*

Uniformly sized space-time cells. (b) Non-uniformly sized space-time cells to accommodate variations in the space-time signature field.

A moving object in a region corresponds to a *peak* in the spatial signal field that moves with time. Tracking an object corresponds to tracking the location of the spatial peak over time. To enable such tracking in a sensor network, the entire space-time region must be divided into *space-time cells* to facilitate local processing, as illustrated in Figure 1. The size of a space-time cell depends on the velocity of the moving target and the decay exponent of the sensing modality. It should approximately correspond to a region over which the space-time signature field remains nearly constant. In principle, the size of space-time cells may be dynamically adjusted as new space-time regions are created based on predicted locations of targets. Space-time signal averaging may be done over nodes in each cell to improve the signal to noise ratio. We note that the assumption of constant signature field over a space-time cell is at best an approximation in practice due to several factors, including variations in terrain, foliage, temperature gradients and non-isotropic nature of source signal. However, such an approximation may be judiciously applied in some scenarios for the purpose of reducing intra-sensor communication as well to improve algorithm performance against noise.

2 Detection and Tracking Framework

In this section, we discuss detection and tracking of a single target in a distributed sensor network. The example illustrates the coordination between networking/routing protocols and CSP algorithms. One of the key premises behind the networking algorithms being developed at Wisconsin is that routing of information in a sensor network should be geographic-centric rather than node-centric [2]. In other words, from the viewpoint of information routing, the geographic locations of the nodes are the critical quantities rather than their arbitrary identities. In the spirit of space-time cells, the geographic region of interest is divided into smaller regions (spatial cells) that facilitate communication over the sensor network. Some of the nodes in each cell are designated as *manager nodes* for coordinating signal processing and communication in that cell.

2.1 Single Target

Figure 2 illustrates the basic idea of region-based CSP for detection and tracking of a single target. Under the assumption that a potential target may enter the monitored area via one of the four corners, four cells, A, B, C and D, are created by the UW-API protocols. Nodes in each of the four cells are activated to detect potential targets.

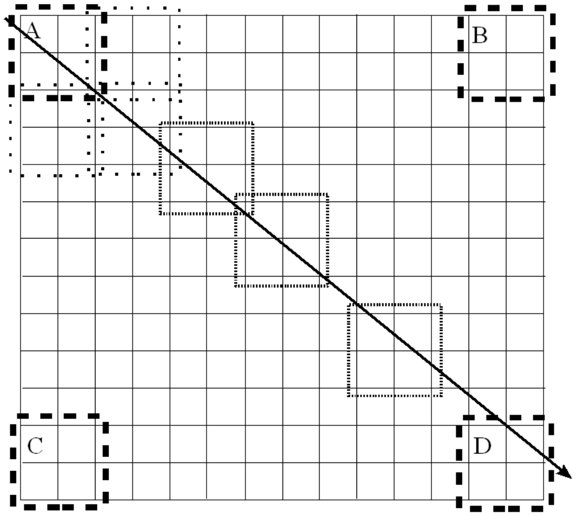

Figure 2. *A schematic illustrating detection and tracking of a single target using UW-API and UW-CSP algorithms.*

Each activated node runs an energy detection algorithm whose output is sampled at an a priori fixed rate depending on the characteristics of expected targets (see Section 2.3.1). Suppose a target enters Cell A. Tracking of the target consists of the following five steps:

(a) Some and perhaps all of the nodes in Cell A detect the target. These nodes are the active nodes and Cell A is the active cell. The active nodes also yield CPA time information. The active nodes report their energy detector outputs to the manager nodes at N successive time instants.

(b) At each time instant, the manager nodes determine the location of the target from the energy detector outputs of the active nodes. The simplest estimate of target location at an instant is the location of the node with the strongest signal at that instant. However, more sophisticated algorithms for target localization may be used (see Section 2.3.2). Such localization algorithms justify their higher complexity only if the accuracy of their location determination is finer than the node spacing.

(c) The manager nodes use locations of the target at the N successive time instants to predict the location of the target at M (< N) future time instants.

(d) The predicted positions of the target are used by the UW-API protocols to create new cells that the target is likely to enter. This is illustrated in Figure 2 where the three dotted cells represent the regions that the target is likely to enter after the current active cell (Cell A in Figure 2). A subset of these cells is activated by the UW-API protocols for subsequent detection and tracking of the target.

(e) Once the target is detected in one of the new cells, it is designated as the new active cell and the nodes in the original active cell (Cell A in Figure 1) may be put in the standby state to conserve energy.

Steps (a) – (e) are repeated for the new active cell and this forms the basis of detecting and tracking a single target. For each detected target, an information field containing tracking information, such as the location of the target at certain past times, is usually passed from one active cell to the next one. This is particularly important in the case of multiple targets. Similar algorithms are being developed by other groups as well [3], [4].

2.2 Multiple Targets

Figure 2 illustrates detection and tracking of a single target. If multiple targets are sufficiently separated in space or time, that is they occupy distinct space-time cells, essentially the same procedure as described in Section 2.1 may be used: a different track is initiated and maintained for each target. Sufficient separation in time means that the energy detector output of a particular sensor exhibits distinguishable peaks corresponding to the CPAs of the two targets. Similarly, sufficient separation in space means that at a given instant the spatial target signatures exhibit distinguishable peaks corresponding to nodes that are closest to the targets at that instant.

The assumption of sufficient separation in space and/or time may be too restrictive in general. In such cases, classification algorithms are needed that operate on spatio-temporal target signatures to classify them. This necessarily requires a priori statistical knowledge of typical signatures for different target classes. In this paper, we focus on single-node (no collaboration between nodes) classification based on temporal target signatures: a time series segment is generated for each detected event at a node and processed for classification. Some form of temporal processing, such as a fast Fourier transform (FFT), is performed and the transformed vector is fed to a bank of classifiers corresponding to different target classes. The outputs of the classifiers that detect the target, active classifiers, are reported to the manager nodes as opposed to the energy detector outputs. Steps (a) – (e) in Section 2.1 are repeated for all the active classifier outputs to generate and maintain tracks for different classified targets. In some cases, both energy-based CPA information and classifier outputs may be needed. The ability to do spatially selective processing at each node via a small antenna array may also be needed in practice [13], [15].

2.3 Signal Processing Algorithms

In this section, we briefly discuss the signal processing algorithms that underlie the tracking framework depicted in Figure 2.

2.3.1 Detection

Energy detection uses minimal a priori information about the target. The detector essentially computes a running average of the signal power over a window of pre-specified length. The output of the detector is sampled at a pre-specified rate. The window duration and sampling rate are determined by target characteristics, such as the signature bandwidth and the expected signature duration in the particular sensing modality. An event is detected when the detector output exceeds a threshold. Due to the inherent signal averaging, the noise component in the output of the detector may be modeled as a Gaussian random variable whose mean and variance can be determined from the statistics of the background noise. The threshold is dynamically adjusted according to the noise variance of detector output so that the detector maintains a constant false alarm rate (CFAR). If the detector output is below the current threshold, the signal is assumed to consist of noise only and these measurements are used to update the threshold.

The output parameters from the energy detector that are communicated to the manager nodes consist of: 1) the onset time when the detector output exceeds the threshold, 2) the time of the maximum (CPA), 3) the detector output at CPA time, and 4) the offset time when the detector output falls below the threshold. For target localization purposes, the detector output at any desired instant within the offset and onset times may also be communicated. For classification purposes, the sensor time series for some fixed duration between the onset and offset times is used.

2.3.2 Target Localization

Determination of target locations at successive time instants is integral to tracking. Various algorithms, with varying levels of complexity, may be developed for this purpose [5], [6], [13], [14], [15]. Here, we briefly outline a simple algorithm for estimating target location at a particular instant by using energy measurements from multiple (4 or more) nodes. Such an energy-based algorithm is a potentially attractive alternative for the following reasons:

(a) A key requirement for accurate localization methods, such as those based on time-delay estimation is accurate synchronization among nodes [7]. However, accurate timing information comes at a rather high cost in wireless sensor networks --- synchronization afforded by relatively cheap hardware may not be accurate enough for such techniques.

(b) Coherent localization methods, such as those using beamforming [6] also require additional assumptions, such as the plane wave (far field) approximation for the incoming wave. Such assumptions are often violated in sensor networks; for example, targets may be rather close to the nodes around CPA time. Alternative methods that obviate such requirements may also be used, albeit at the cost of requiring more statistical information (see, e.g., [5], [14], [15]).

(c) Exchange of time series data among nodes, as required by some algorithms, consumes too much energy to be feasible.

Our energy-based target localization algorithm assumes an isotropic exponential attenuation for the target energy source: $y_i(t) = s(t) / \|\mathbf{r}(t) - \mathbf{r}_i\|^{\alpha}$, where $y_i(t)$ is the energy reading at the i^{th} sensor, $\mathbf{r}(t)$ denotes the *unknown* coordinates of the source with respect to a fixed reference, $\mathbf{r}_i$ are the coordinates of i^{th} sensor, s(t) is the *unknown* target signal energy, and α is the decay exponent which is assumed to be known (or estimated via experiments [20]). The algorithm first computes the ratios $y_i(t)/y_j(t)$ for all pairs of sensors to eliminate the unknown variable s(t). Each ratio defines a circle on which $\mathbf{r}(t)$ may reside. In the absence of noise, it can be shown that for n sensor readings only n-1 of the total n(n-1)/2 ratios are independent, and all the corresponding circles intersect at a single point for 4 or more sensor readings. For noisy measurements, more than n-1 ratios may be used for robustness, and the unknown target location $\mathbf{r}(t)$ is estimated by solving a nonlinear least squares problem of the form $J(x,y) = \sum_{i=1}^{m} \left\| (x - o_{i,x})^2 + (y - o_{i,y})^2 - \rho_i^2 \right\|^2$, where (x,y) are the unknown coordinates of the target, $(o_{i,x}, o_{i,y})$ are the center coordinates, and ρ_i the radius of the circle associated with the i^{th} ratio. A sample contour plot for the estimated target location determined by the algorithm is shown in Figure 3 for seismic time series. Accuracy of the position estimates depends on the accuracy with which the node locations and the attenuation exponent can be measured. A detailed sensitivity analysis is beyond the scope of this paper.

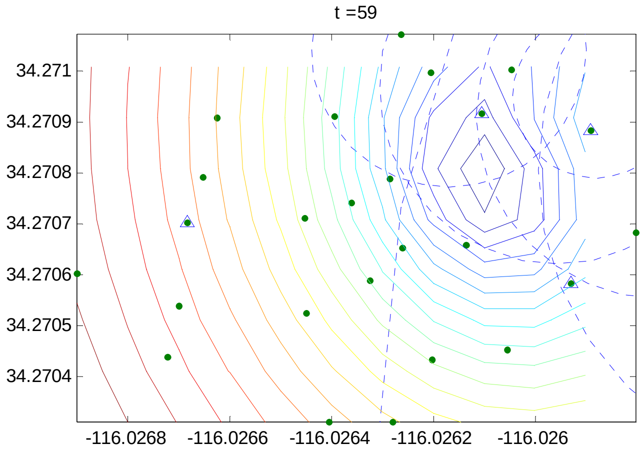

Figure 3. *Illustration of energy based collaborative target localization using real seismic (noisy) data. Dots indicate node locations. Triangles indicate the 4 nodes being used in the position calculation. Dashed circles indicate loci of possible target locations based on the 6 ratios of sensor readings. Three different points can be identified at which 3 of the 6 circles intersect, suggesting possible target locations. Solid lines represent constant-value contours of the cost function. The center of the contour plots indicates estimated target location and is inside the region bounded by the 3 intersection points of circles.*

2.3.3 Target Tracking

Given target locations at time instants in the past, it is possible to fit the data samples into a dynamic model to predict future target locations [3], [4]. For a single moving target, sufficiently accurate tracking may be accomplished by fitting the data into a linear or polynomial model using a least square fit. Tracking is a complicated problem when multiple targets are present; targets tracks can cross paths resulting in the data-association problem. Classification algorithms can provide a solution.

2.3.4 Remarks

We note that many of the assumptions made in the above sections may be violated in practice making the problem of target localization and tracking much more difficult. For example, the propagation decay constant may vary between sources of same modality. The signal strength may also be a function of direction depending on the location of the source within the target. Furthermore, there is interference between different targets in the case of multiple targets. Classification algorithms, in principle, should suppress interference from undesired sources. Small antenna arrays at each node to enable spatially selective processing could also be extremely beneficial.

3 Target Classification

As mentioned earlier, classification algorithms are needed in general for tracking multiple targets. Classification algorithms operate on time-series data associated with each detected event. Collaborative classification algorithms may operate on *space-time* signatures associated with each event over a number of nodes in a space-time cell. Furthermore, space-time signatures corresponding to multiple modalities, such as, acoustic and seismic, may be combined for improved performance [8]. However, such collaborative classification puts a very significant burden on the network since time-series data from multiple nodes has to be transmitted to the manager nodes.

In this paper, we focus on single-node classification algorithms that operate on time series segments associated with detected events. As we discuss in Section 4, the variability in temporal signatures from event to event for a particular target class poses a significant challenge in classifier design. In general, some a priori knowledge of the statistical characteristics of signatures for different target classes is required [8]. Some aspects of signature variability may be accounted for deterministically via nuisance parameters [11], [12], as briefly discussed in Section 5.3. While our focus is on single-node classification algorithms as a logical first step, collaborative processing may be necessary in practice for desired performance and we discuss such issues in Section 5.2.

In this section, we explore the performance of three classification algorithms: k-nearest neighbor (kNN) classifier, maximum likelihood (ML) classifier using Gaussian data modeling, and Support Vector Machine (SVM) classifier. The emphasis is on results obtained from real seismic and acoustic data collected in SITEX00 experiments performed in the DARPA SenseIT program. We report results for binary classification between *wheeled* and *tracked* vehicles.

3.1 Spectral Features

The choice of feature vectors on which the classifiers operate is critically important to their performance. The experiments with real data discussed in this section correspond to moving vehicles. Due to the dominant effect of rotating machinery (engine, gears, wheels, etc.) as well other periodic phenomena, such as tread-road impact, on acoustic and seismic signatures, *spectra* of time series suggest themselves as useful features. Our initial experimentation has shown that spectral characteristics of signatures vary significantly between target classes and hence may be fruitful for classification.

We have explored two types of spectral features: 1) Non-parametric FFT-based power spectral density (PSD) estimates, and 2) Parametric PSD estimates using autoregressive (AR) modeling of time series. We explored AR modeling (with fewer parameters) primarily to improve the statistical reliability of the PSD estimates. However, due to space limitations, we report results based only on non-parametric PSD features. In our experiments to date, the performance with AR spectral features is slightly inferior.

Figure 4 shows the seismic PSDs for three different vehicles (one tracked and two wheeled). The sampling rate is 256Hz and the detected events consist of 3000, 2500, and 2000 samples for events 08020830, 08030800, and 08040820, respectively. For each event, PSD estimates are based on averaging length 256 FFTs of 0.25 second (64 samples) zero-padded data segments, with an overlap of 32 samples between adjacent data segments. Only the positive 128 FFT points are plotted. As evident, the plots of the three vehicles show different dominant frequencies that may be exploited by classifiers. We note that additional time-averaging may be done to reduce the variance of PSD estimates by using shorter FFT segments, however this comes at the cost of smoothing spectral features (bias-variance tradeoff).

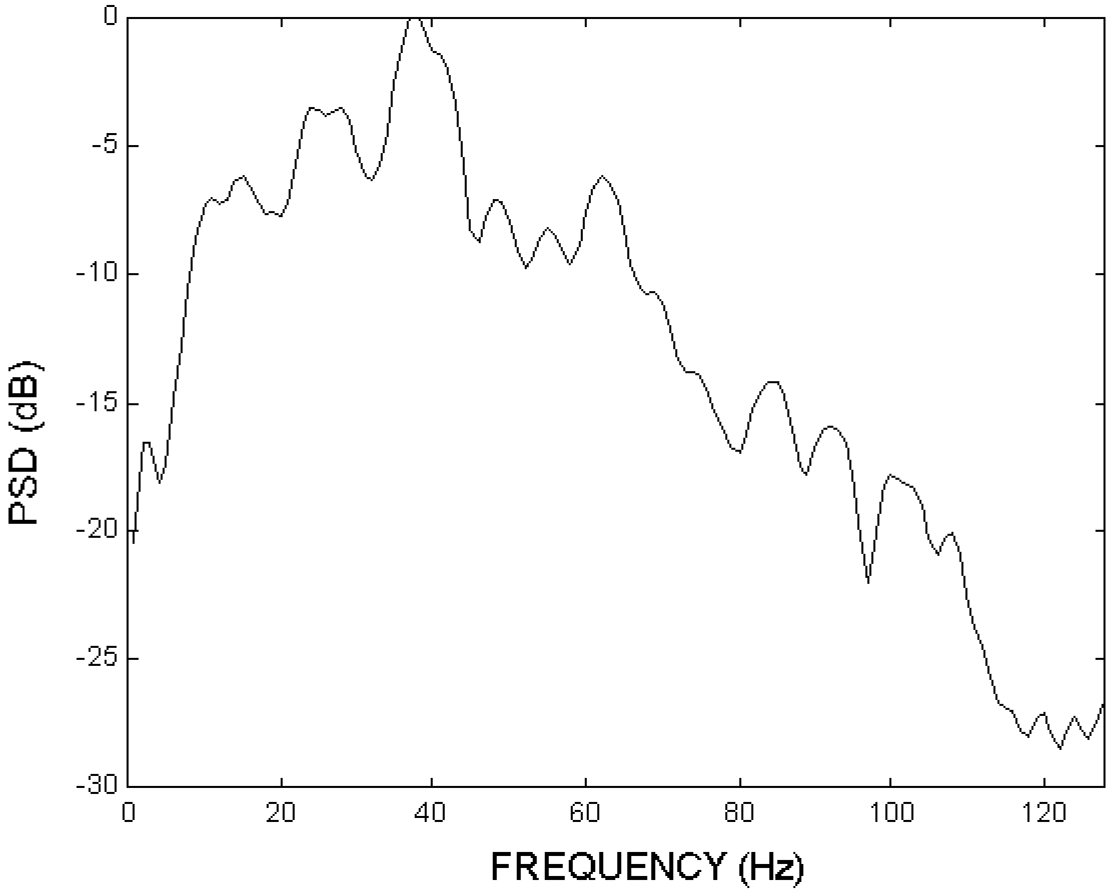

Figure 4a: *Seismic PSD of a wheeled vehicle (08020830 DW)*

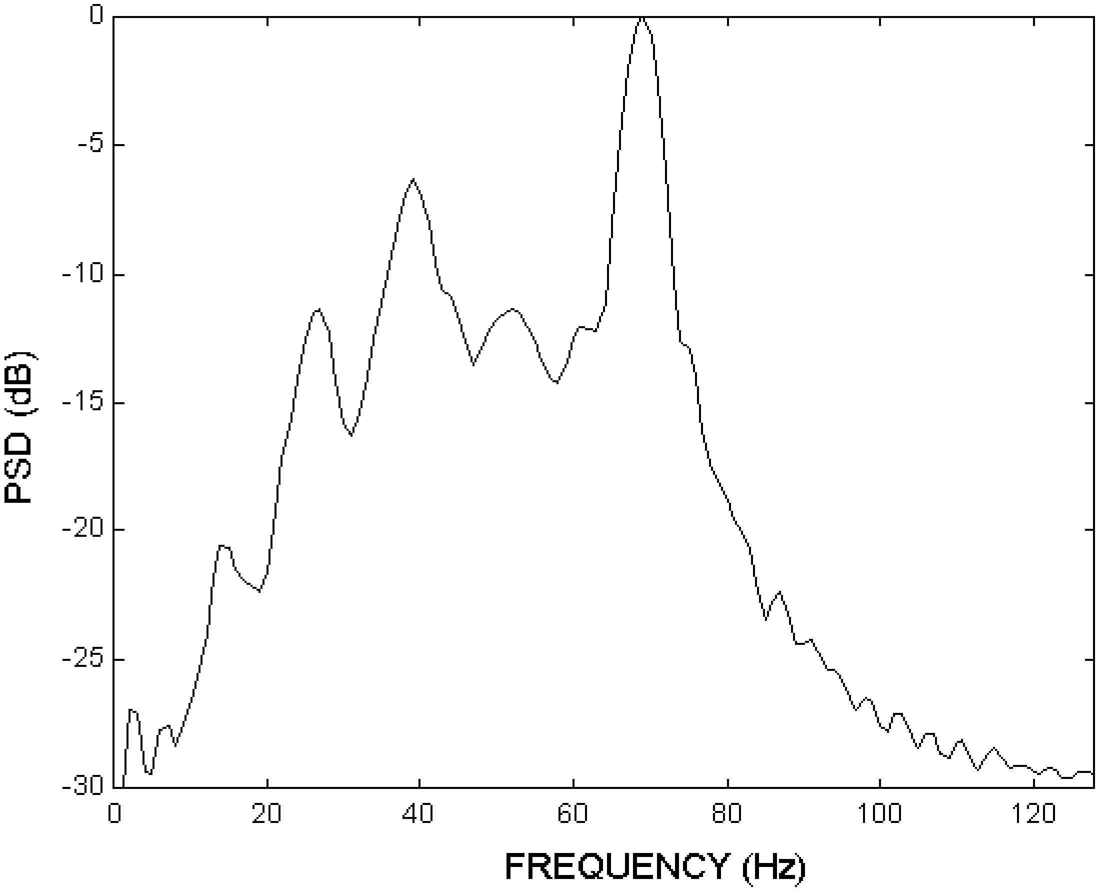

Figure 4b: *Seismic PSD of a tracked vehicle (08030800 AAV)*

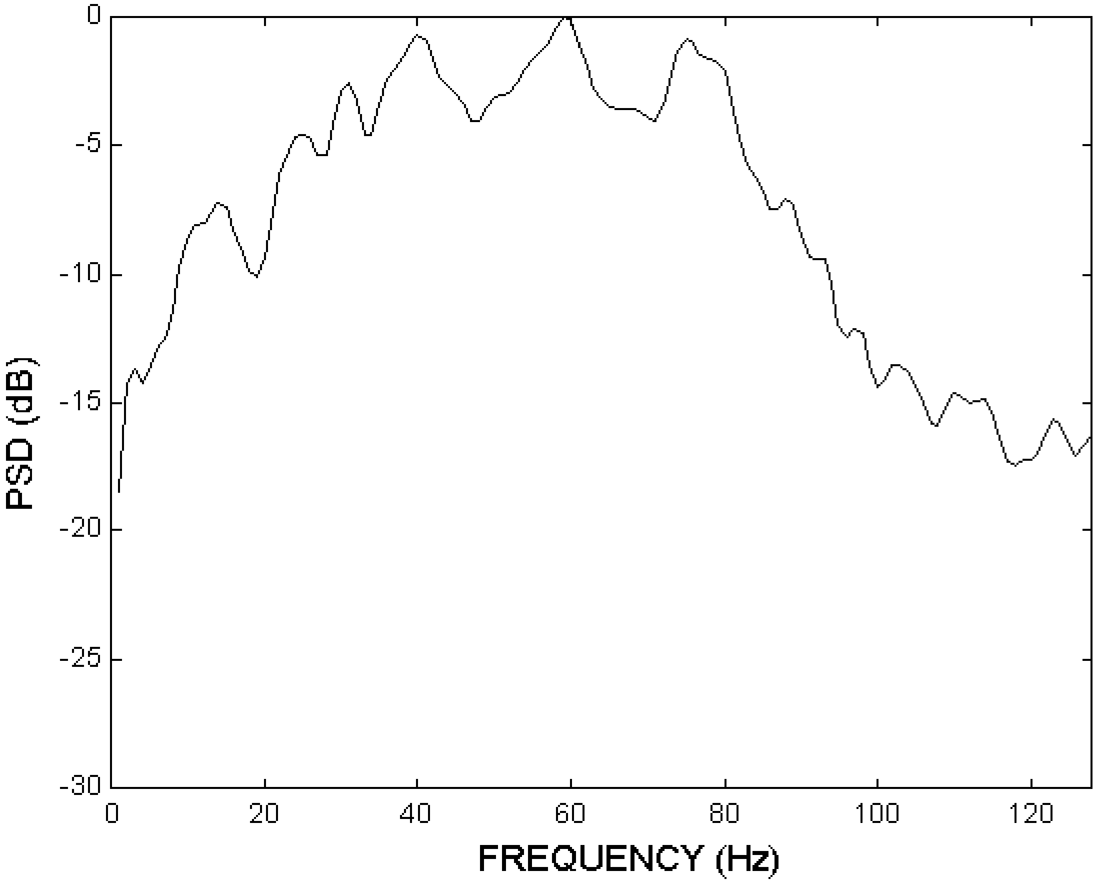

Figure 4c: *Seismic PSD of a wheeled vehicle (08040820 LAV)*

Figure 5 shows PSDs for the same three vehicles as in Figure 4 using wideband acoustic data collected by BAE and Xerox. The data is downsampled to 1024 Hz and the PSDs are generated by averaging length-1024 FFTs of 0.25 second (256 samples) data segments. Only the positive 512 frequency points are plotted. It is worth noting that the spectra from the two data sets show different dominant spectral peaks for the wheeled vehicles (Figures 5a and 5c), even though they supposedly correspond to the same event involving the same vehicle. This illustrates one of the challenges associated with classification --- *variability in measurements* --- discussed in more detail in Section 4.

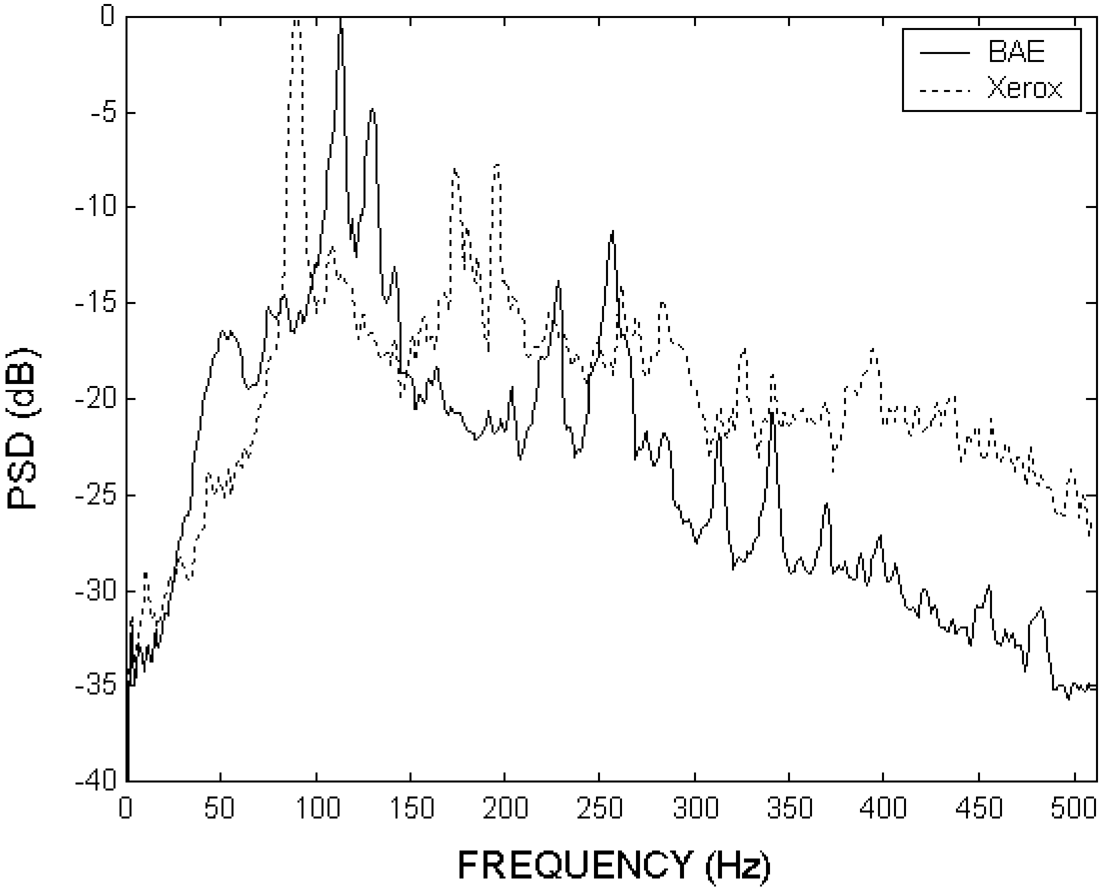

Figure 5a: *Acoustic PSD of a wheeled vehicle (08020830 DW)*

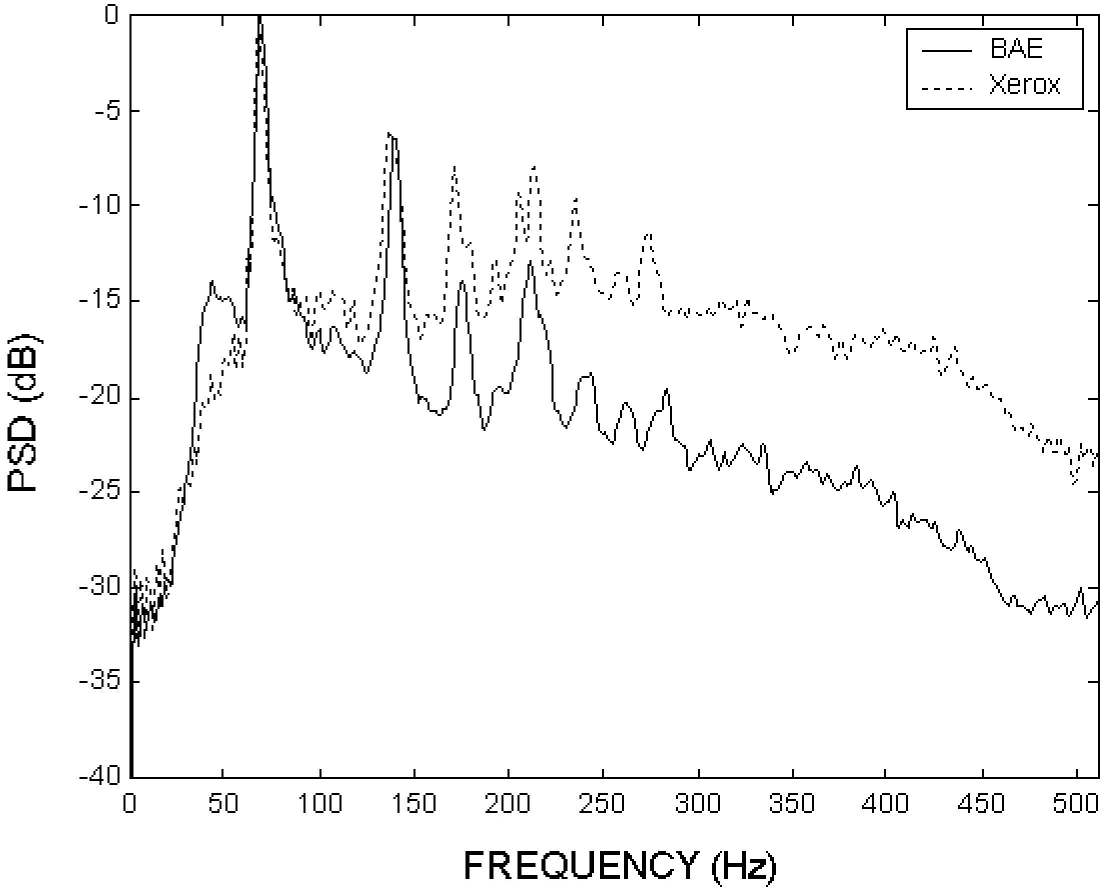

Figure 5b: *Acoustic PSD of a tracked vehicle (08030800 AAV)*

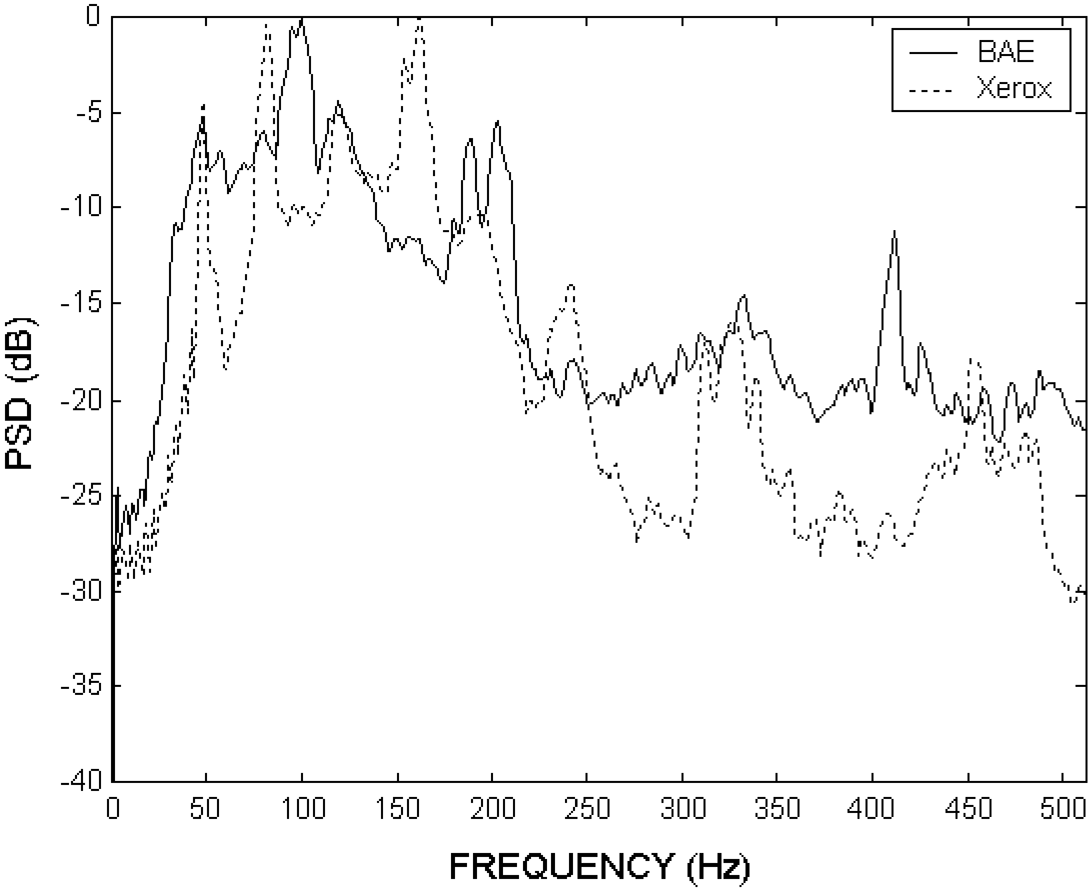

Figure 5c: *Acoustic PSD of a wheeled vehicle (08040820 LAV)*

3.2 Classification Algorithms

In this section we briefly describe the three classifiers explored in this paper. Given a set of N-dimensional feature vectors {**x**; **x** ∈ $\Re^N$}, we assume that each of them is assigned a class label, $\omega_c \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$, that belongs to a set of m elements. We denote by $p(\omega_c)$ the *prior probability* that a feature vector belongs to class ω_c. Similarly, $p(\omega_c|\mathbf{x})$ is the *posterior probability* for class ω_c given that **x** is observed.

A *minimum error classifier* maps each vector **x** to an element in *Ω* such that the probability of misclassification --- the probability that the classifier label is different from the true label --- is minimized. To achieve this minimum error rate, the optimal classifier decides **x** has label ω_i if $p(\omega_i|\mathbf{x}) > p(\omega_j|\mathbf{x})$ for all $j \neq i$, ω_i, $\omega_j \in \Omega$. In practice, it is very difficult to evaluate the posterior probability in closed form. Instead, one may use an appropriate *discriminant function* $g_i(\mathbf{x})$ that satisfies $g_i(\mathbf{x}) > g_j(\mathbf{x})$ if $p(\omega_i|\mathbf{x}) > p(\omega_j|\mathbf{x})$ for $j \neq i$, for all **x**. Then minimum error classification can be achieved as: decide **x** has label ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for $j \neq i$. The minimum probability of misclassification is also known as the Bayes error, and a minimum error classifier is also known as a Bayes classifier or a maximum a posterior probability (MAP) classifier. Below, we briefly discuss three classifiers that approximate the optimal Bayes classifier.

3.2.1 k-Nearest Neighbor (kNN) Classifier

The kNN classifier uses *all* the training features as the set of *prototypes* $\{\mathbf{p}_k\}$. During testing phase, the distance between each test vector and every prototype is calculated, and the k prototype vectors that are closest to the test vector are identified. The class labels of these k-nearest prototype vectors are then combined using majority vote or some other method to decide the class label of the test vector. When k=1, the kNN classifier is called the nearest neighbor classifier. It is well-known [9] that asymptotically (in the number of training vectors), the probability of misclassification of a nearest neighbor classifier approaches twice the (optimal) Bayes error. Hence the performance of a nearest neighbor classifier can be used as a baseline to gauge the performance of other classifiers. However, as the number of prototypes increases, a kNN classifier is not very suitable for actual implementation since it requires too much memory storage and processing power for testing.

3.2.2 Maximum Likelihood Classifier Using Gaussian Mixture Density Model

In this classifier, the distribution of training vectors from the same class is modeled as a mixture of Gaussian density functions. That is, the *likelihood function* is modeled as

$$p(\mathbf{x}|\omega_i) \propto G_i(\mathbf{x}|\boldsymbol{\theta}_i) = \sum_k |\boldsymbol{\Lambda}_{ik}|^{-N/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m}_{ik})^T \boldsymbol{\Lambda}_{ik}{}^{-1}(\mathbf{x}-\mathbf{m}_{ik})\right)$$

where $\boldsymbol{\theta}_i = [\mathbf{m}_{i1}, \ldots, \mathbf{m}_{iP}, \boldsymbol{\Lambda}_{i1}, \ldots, \boldsymbol{\Lambda}_{iP}]$ are the mean and covariance matrix parameters of the P mixture densities corresponding to class ω_i. These model parameters can be identified by applying an appropriate clustering algorithm, such as the *k-means algorithm* [9], or the Expectation-Maximization algorithm [28] to the training vectors of each class. The discriminant function is computed as $g_i(\mathbf{x}) = G_i(\mathbf{x}|\boldsymbol{\theta}_i)p(\omega_i)$ where the prior probability $p(\omega_i)$ is approximated by the relative number of training vectors in class i. In the numerical examples we model the data as Gaussian rather than a Gaussian mixture (P=1). Furthermore, we use the maximum likelihood (ML) classifier (uniform prior probabilities).

3.2.3 Support Vector Machine (SVM) Classifier

A support vector machine (SVM) is essentially a linear classifier operating in a higher dimensional space. Consider a binary classification problem without loss of generality. Let $\{\varphi_i(\mathbf{x})\}_{i=1}^M$ be a set of nonlinear transformations mapping the N-dimensional input vector to an M-dimensional feature space (M > N). A linear classifier, characterized by the weights $\{w_1, w_2, \ldots, w_M\}$, operates in this higher dimensional feature space $g(\mathbf{x}) = \sum_{j=1}^{M} w_j \varphi_j(\mathbf{x}) + b$, where b is the bias parameter of the classifier [10]. The optimal weight vectors for this classifier can be represented in terms of a subset of training vectors, termed the *support vectors* $w_j = \sum_{i=1}^{Q} \alpha_i \varphi_j(\mathbf{x}_i)$, j = 1, 2, …, M. Using the above representation for the weight vectors, the linear classifier can be expressed as $g(\mathbf{x}) = \sum_{i=1}^{Q} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b$, where $K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^{M} \varphi_j(\mathbf{x}) \varphi_j(\mathbf{x}_i)$ is the symmetric kernel representing the SVM. In the numerical examples presented in this paper, we use a third degree polynomial kernel: $K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^3$. In practice, the SVM discriminant function g(**x**) is computed using the kernel representation, bypassing the nonlinear transformation into the higher dimensional space [10]. The classifier design then corresponds to the choice of the kernel and the support vectors. By appropriately choosing the kernel, an SVM can realize a neural network classifier as well. Similar to neural networks, the training phase can take a long time. However, once the classifier is trained, its application is relatively easy. In general, a different SVM is trained for each class. The output of each SVM can then be regarded as an estimate of the posterior probability for that class and the MAP decision rule can be directly applied.

3.3 Performance Assessment

3.3.1 Cross Validation

Performance assessment is a critical component of classifier design. In practice, the probability of misclassification must be evaluated empirically. The available feature vectors are divided into two sets: a *training set* that is used to train the classifier, and a *testing set* that is used to estimate its probability of misclassification. In the numerical results presented in the following sections, we use 3-way cross validation for performance assessment. The data is divided into three sets (F1, F2, F3) and three different sets of experiments are performed for performance assessment:

A	B	C
Training: F1 + F2; Testing: F3	Training: F2 + F3; Testing: F1	Training: F3 + F1; Testing: F2

The results are reported for binary classification between wheeled and tracked vehicles using both low-bandwidth seismic and wideband acoustic data.

3.3.2 Low Bandwidth Seismic Data

We compare the performance of the three classifiers on low-bandwidth data from seismic sensors collected in the SITEX00 experiments. The sensor outputs were sampled at 256 Hz. The spectral feature vectors consist of the positive 64 samples of 128 length FFT corresponding to 0.5 second time series segments. There were two types of tracked vehicles and five types of wheeled vehicles in the data set. The three data sets for 3-way cross validation had the following numbers of features in each:

	F1	F2	F3	Total
Tracked	73	73	74	220
Wheeled	166	166	168	500
Total	239	239	242	720

The confusion matrices for the three classifiers are tabulated below. Vertical labels correspond to true labels whereas the horizontal labels correspond to the classifier labels.

K-Nearest Neighbor (K = 1)

	Tracked	Wheeled
Tracked	197 (89.55%)	23 (10.45%)
Wheeled	24 (4.80%)	476 (95.2%)

Maximum Likelihood (Gaussian Modeling)

	Tracked	Wheeled
Tracked	203 (92.27%)	17 (7.73%)
Wheeled	112 (22.4%)	388 (77.6%)

SVM

	Tracked	Wheeled

Tracked	207 (94.09%)	13 (5.91%)
Wheeled	15 (3.0%)	485 (97.0%)

As evident, the nearest neighbor and SVM classifiers perform quite well. The somewhat inferior performance of the ML classifier may be due to the fact that we do not account for prior class probabilities which are implicitly accounted for in the kNN and SVM classifiers. (Note that there are about twice as many wheeled feature vectors as tracked ones.) We note that similar performance has been reported by other researchers in related contexts using wavelet-based processing [18], [19].

3.3.3 Wideband Acoustic Data

During SITEX00 experiments, research groups from BAE and Xerox PARC both recorded broadband acoustic data. The BAE data was sampled at 10 KHz and the Xerox data was sampled at 20 or 40 KHz. Both data sets were down-sampled to 5 KHz before applying the classification algorithms. Based on videotapes recorded during the experiment, each run (consisting of several events) was segmented into 5-second segments and each segment was manually labeled as tracked or wheeled (more accurate ground truth information was not available).

For each labeled 5-second acoustic clip (sampled at 5 KHz), a 1000-point FFT is performed for each non-overlapping segment of 1000 samples. This yields 25 1000-point FFT vectors for each 5-second segment. These vectors are then magnitude squared to yield PSD estimates at a resolution of 5 Hz per spectral bin. Our study of the data indicated that most of the signal energy is confined to frequencies below 1 KHz. Thus, we used the Fourier coefficients corresponding to the first 200 positive spectral bins. To further reduce the feature dimension and to improve statistical reliability, we extracted two types of spectral features from these non-parametric PSD estimates:

Feature type 1. Non-overlapping sets of 4 adjacent spectral bins are averaged to yield a 50 by 1 feature vector with each entry representing the signal energy in a frequency band of 20 (5×4) Hz.

Feature type 2. Non-overlapping sets of 10 adjacent spectral bins are averaged to yield a 20 by 1 feature vector with each entry representing the signal energy in a frequency band of 50 (5×10) Hz.

Each feature vector is then level-shifted to make it average to zero and is normalized using the L_∞ norm.

The three data sets used in 3-way cross validation had the following number of features in each class:

	F1	F2	F3	Total
Tracked	324	306	329	959
Wheeled	512	530	508	1550
Total	836	836	837	2509

The confusion matrices for the three classifiers are reported below**.**

K-Nearest Neighbor (K = 1)

	Tracked	Wheeled
Tracked	842 (87.80%)	117 (12.20%)
Wheeled	89 (5.74%)	1461 (94.26%)

Maximum Likelihood (Gaussian Modeling)

	Tracked	Wheeled

Tracked	779 (81.23%)	180 (18.77%)
Wheeled	171 (11.03%)	1379 (88.97%)

SVM

	Tracked	Wheeled
Tracked	887 (92.50%)	72 (7.5%)
Wheeled	55 (3.55%)	1495 (96.45%)

The SVM classifier again performs the best and ML classifier the worst. The performance of the ML classifier could be better if the prior class probabilities reflected the different number of vehicles in the two classes.

4 Issues and Challenges

Making CSP work in real sensor networks presents many hurdles. In this section, we describe some of the issues and challenges that we have encountered and identified in our experience with real data. In general the issues stem from some form of *variability in measurements*. We focus on three major sources of variability: 1) Uncertainty in space-time measurements, 2) Doppler effects due to motion, and 3) variability in experimental conditions. Such variability is particularly challenging in classification algorithms since they rely on statistical signal characteristics inferred from training data --- targets may exhibit significantly different characteristics during the test experiments as compared to the training phase. Making the classifiers robust to such variability is critical for their successful application and is a very challenging task.

4.1 Uncertainty in Temporal and Spatial Measurements

Accurate timing synchronization between nodes and accurate estimates of their positions is critical to many CSP algorithms. For example, the accuracy of the target localization algorithm described in Section 2.3.2, and that of related algorithms for estimating target bearing [7], critically depends on such timing and location information. Similarly, relative locations of nodes are exploited for estimating time difference of arrival between nodes in bearing estimation and beamforming. Errors in node locations can lead to errors in timing information in such cases. Sufficiently accurate timing and spatial information may not be available even when the sensor nodes are equipped with GPS receivers. Signal averaging within spatial cells may be used to improve the accuracy of timing and location information.

4.2 Effect of Doppler on Spectral Signatures

An implicit assumption in the training of the classifiers discussed in Section 3 is that statistical characteristics of the target signatures do not change over the duration of the detected event. This assumption may not hold in practice. For example, gear shifts, acceleration and simply Doppler shifts due to motion can result in significantly varying spectral characteristics. Doppler effects in acoustic and seismic signals are particularly significant due to relatively low speed of wave propagation.

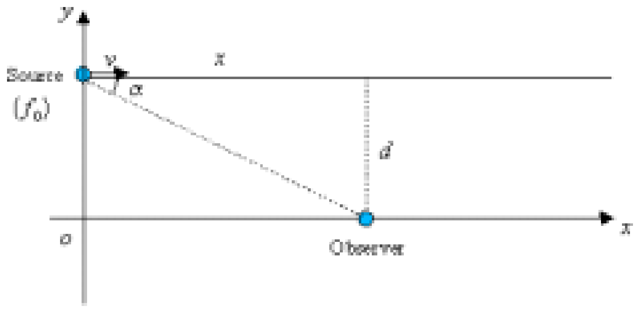

Figure 6. *A simple geometry for a moving source to illustrate Doppler effects.*

Consider the set up in Figure 6 to illustrate Doppler effects. A source emitting energy at frequency f_o is moving at velocity ν parallel to the x-axis. The perpendicular distance between the source and the observer (sensor) is d. The sensor is located at a distance x along the axis. A simple calculation shows that the frequency *perceived* by the sensor is related to source frequency as $f = \frac{f_0}{1-(v/v_0)\cos\alpha}$ where α is the angle between the x-axis and the line-of-sight between the source and the sensor.

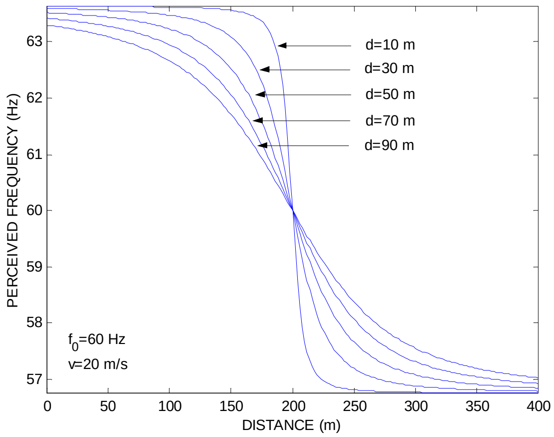

Figure 7. *Plots of perceived frequency for acoustic signals as a function of source position for different values of d.*

Figure 7 plots the perceived frequency for acoustic data as a function of source position for different values of d. The source frequency is f_o = 60 Hz, the sensor is located at $x = 200$ m, and the source velocity is 20m/s. It is evident that the perceived frequency is equal to the source frequency at the CPA and variation in the perceived frequency gets sharper for smaller d. Similarly, the total frequency variation increases with speed. Actual data from experiments confirms similar spectral trends. Figures 8 and 9 show the short-time Fourier transform (STFT) plots of seismic and acoustic signals, respectively, for a vehicle moving past a node. The variation in perceived frequency, similar to that in Figure 7, is evident. Note that the seismic signature is shorter in time due to faster signal decay with distance in ground. The perceived frequency variation in acoustic data could also be partly due to gear shifts. The important thing is that the changes in perceived frequency are significant at normal source speeds and thus must be taken into account. One possibility is to use the ratio of dominant frequencies as a feature to make classification algorithms robust to Doppler shifts.

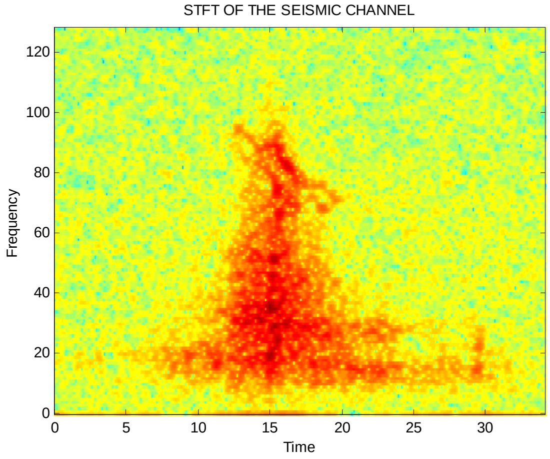

Figure 8. *STFT of a seismic signal due to a moving vehicle.*

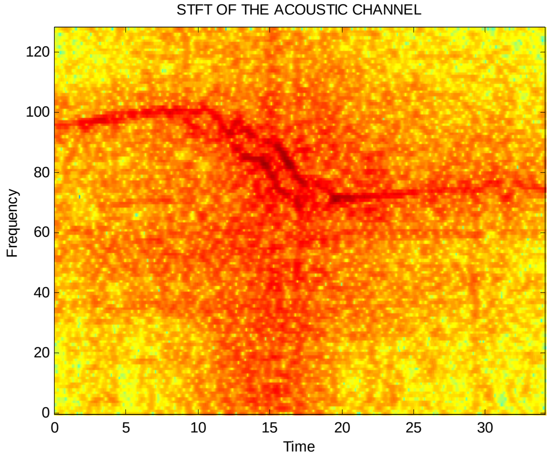

Figure 9. *STFT of an acoustic signal due to a moving vehicle.*

4.3 Variability in Experimental Conditions

Another, and more stringent, assumption in the classification algorithms discussed in Section 3 is that the signal characteristics for a particular target class are relatively consistent between different events involving the same target. This assumption is needed since data from prior experiments is used to train the classifiers for future experiments. However, many factors in practice violate this assumption, including varying conditions between different experiments, such as the node layout. For example, nodes closer to roads will experience sharper Doppler shifts than farther nodes, as illustrated in Figure 7. This variability may be the cause of differences in Xerox and BAE spectral signatures for the same vehicle, as illustrated in Figures 5a and 5c. The net effect of such variability is that each target class occupies more signal space dimensions, thereby requiring a sufficiently high dimensional signal representation for reliable classification. This may be another reason for the superior performance of the SVM classifier. Work by other researchers on the calibration of unattended sensors may be useful in attacking this problem [21].

5 Future Research

We now briefly identify some avenues for future research in CSP for sensor networks based on our investigations and experiences with real data. Recall that of the algorithms discussed in this paper, only target localization and location prediction require collaborative processing. Detection and classification algorithms have been described for single-node processing. However, given the myriad of effects that can compromise the performance of signal processing algorithms in a sensor network, various forms of CSP may be necessary. As in all cases, there is a performance versus complexity tradeoff that is particularly acute in sensor networks since collaboration between nodes comes at the cost of exchanging information between them.

5.1 Intra-sensor Collaboration: Modal Fusion

One of the simplest and most feasible forms of CSP is to combine the information across different sensing modalities at each node. The information to be combined resides at one node and thus does not require communication over the network. For example, jointly processing acoustic and seismic signals may significantly enhance performance. Higher dimensional feature vectors obtained by concatenating the vectors for the two modalities may be used for classification. A lower complexity alternative may be to combine the *decision* statistics for different modalities.

5.2 Inter-sensor Collaboration: Centralized Processing

Another possibility for CSP is to effectively perform joint space-time processing across different nodes in a cell via manager nodes. Consider classification for illustration. All active nodes in a cell send the time series data corresponding to the detected event to the manager node. The manager node then jointly processes the time series data to classify the target. *Temporally coherent* processing would be possible when the relative timing of the different time series is known and exploited, such as in time-of-arrival based bearing estimation algorithms [7]. Similarly, *spatially coherent* processing may be possible if the relative locations of nodes are known and exploited. For example, distributed spatial beamforming may be effectively implemented in such cases to localize targets as well as to suppress interference from unwanted targets. Significant research in array processing and more recently in space-time processing for wireless communications could be fruitfully exploited in this context (see, e.g., [13],[26],[27]).

5.3 Doppler Based Composite Hypothesis Testing

Doppler shifts represent one important source of variability that can be explicitly modeled, at least for simple scenarios, as discussed in Section 4.2. We briefly discuss one approach based on a generalized likelihood ratio test (GLRT) [11]. In the presence of Doppler shifts, the target signature for the mth class can be modeled as a realization of a zero-mean wide-sense stationary process modulated by a time-varying instantaneous frequency, $f(\boldsymbol{\theta}_m)$, parameterized by the vector, $\boldsymbol{\theta}_m = [\alpha_m \quad v_m \quad d_m]$, corresponding to the scenario in Figure 6. A GLRT classifier essentially operates as follows: First an ML estimate of the Doppler parameters, $\hat{\boldsymbol{\theta}}_m$, is obtained for each hypothesis and then the posterior probability density functions corresponding to these estimates, $p(\omega_m \mid \mathbf{x}, \hat{\boldsymbol{\theta}}_m)$, are used for deciding between the classes. Such a GLRT-based approach requires "alignment" of data to some fixed frequency during the training phase, which may be done in a variety of ways [25]. The aligned data may then be used to train any one of the three classifiers. Clearly, such algorithms require more computation and the performance-complexity tradeoff has to be carefully evaluated.

6 Conclusions

In this paper we have presented the basic ideas behind a CSP framework for tracking multiple targets in a distributed sensor network. The key components of the framework include event detection, estimation and prediction of target location, and target classification. Most of the existing work is for tracking a single target and is based on a single sensing modality, such as acoustic or seismic. Tracking of multiple targets in general requires classification algorithms. Based on experimentation with real data, we have argued that spectral target signatures may yield accurate classification. However, variations in spectral signatures due to a variety of effects, including Doppler shifts, presents a significant challenge. We have provided some promising preliminary results on classifying between wheeled and tracked vehicles. The SVM classifier seems to be the most promising based on our initial experiments.

The algorithms discussed in this paper primarily represent a snapshot of the ongoing work at Wisconsin on the extremely challenging problem of tracking multiple targets in a sensor network and are by no means the best approach. We are also working on incorporating insights and results obtained by other researchers. Results from this ongoing project, including code for implementing various algorithms, will be posted on our website **http://www.ece.wisc.edu/~sensit/.**

In closing, distributed sensor networks are emerging as a powerful technology for remotely instrumenting and monitoring the physical world. However, the technology is in its infancy and many challenges need to be overcome before it becomes practically feasible. Several research groups around the country are tackling important pieces of the puzzle. The results and findings have been very promising and are leading to new research activity to advance the theory and practice of sensor networks.

Acknowledgement

This work was partly supported by DARPA under grant no. F 30602-00-2-0555. The authors would like to thank Steve Beck, Peter Boettcher, Richard Brooks, David Friedlander, Jim Reich, Joe Reynolds, and Ken Theriault for their useful comments during the course of this work. The authors also gratefully acknowledge the useful comments of one of the reviewers.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, “Instrumenting the world with wireless sensor network,” *Proc. ICASSP'2001*, Salt Lake City, UT, pp. 2675-2678, 2001.

[2] P. Ramanathan, K. Saluja, K-C. Wang, T. Clouqueur, "UW-API: A Network Routing Application Programmer's Interface", draft version 1.0, January 2001.

[3] R. Brooks, C. Griffin, and D. S. Friedlander, "Self-Organized distributed sensor network entity tracking," to be published in the *International Journal of High Performance Computing Applications*, Special Issue on Sensor Networks, 2002.

[4] R. Brooks and C. Griffin, "Traffic model evaluation of ad hoc target tracking algorithms," to be published in the *International Journal of High Performance Computing Applications*, Special Issue on Sensor Networks, 2002.

[5] J. C. Chen, R. E. Hudson, and K. Yao, “A Maximum likelihood parametric approach to source localization,” *Proc. ICASSP'2001*, Salt Lake City, UT, pp. 1043-1046, 2001.

[6] L. M. Kaplan, Q. Le, and P. Molnar, “Maximum likelihood methods for bearings-only target localization,” *Proc. ICASSP'2001*, Salt Lake City, UT, pp. 554-557, 2001.

[7] P. W. Boettcher and G. A. Shaw, "A Distributed Time-Difference of Arrival Algorithm for Acoustic Bearing Estimation", *Proc. 4th International Conf. on Information Fusion*, vol. 1, Montreal, August 2001.

[8] D. S. Friedlander and S. Phoha, "Semantic Information Fusion of Coordinated Signal Processing in Mobile Sensor Networks," to be published in *The International Journal of High Performance Computing Applications*, Special Issue on Sensor Networks, 2002.

[9] R. O. Duda and P. E. Hart, Pattern classification and Scene Analysis, New York, Wiley, 1973.

[10] S. Haykin, *Neural Networks*, Prentice Hall

[11] H. V. Poor, *An Introduction to Signal Detection and Estimation*, Springer Verlag, 1988.

[12] A. M. Sayeed and D. L. Jones, "Optimal Detection Using Bilinear Time-Frequency and Time-Scale Representations," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2872-2883, December, 1995.

[13] K. Yao, R. E. Hudson, C. W. Reed, D. Chen, and F. Lorenzelli, "Blind Beamforming on a Randomly Distributed Sensor Array System," *IEEE Journal on Selected Areas in Communications*, Special Issue on Signal Processing for Wireless Communications, vol. 16, no. 8, pp. 1555-1567, Oct. 1998.

[14] C. W. Reed, R. Hudson, and K. Yao, "Direct Joint Source Localization and Propagation Speed Estimation," *Proc. IEEE ICASSP*, March 1999, pp. 1169-1172.

[15] K. Yao, R. E. Hudson, C. W. Reed, T. L. Tung, D. Chen, and J. C. Chen, "Estimation and Tracking of an Acoustic/Seismic Source Using a Beamforming Arrays based on Residual Minimizing Methods," *Proc. 1999 Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing*, Sept. 1999, pp. 129-136.

[16] J. R. Agre and L. P. Clare, "An Integrated Architecture for Cooperative Sensing Networks," *Computer*, vol. 33, May 2000, pp. 106-108.

[17] L. P Clare, G. J. Pottie, and J. R. Agre, "Self-Organizing Distributed Sensor Networks," *Proc. SPIE*, vol. 3713, April 199, pp. 229-238.

[18] J. F. Scholl, J. R. Agre, L. P. Clare, and M. C. Gill, "A Low Power Impulse Signal Classifier Using the Haar Wavelet Transform," *Proc. SPIE*, vol. 3577, Nov. 1998, pp. 136-145.

[19] J. F. Scholl, J. R. Agre, and L. P. Clare, "Wavelet Packet Target Classification Schemes," *Proc. 1999 Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing*, Sept. 1999 (17 pages).

[20] J. F. Scholl, L. P. Clare, and J. R. Agre, "Seismic Attenuation Characterization Using Tracked Vehicles," *Proc. 1999 Meeting of the MSS Specialty Group on Battlefield Acoustic and Seismic Sensing*, Sept. 1999 (11 pages).

[21] R. Moses, R. Patterson, D. Krishnamurthy, N. Srour, and T. Pham, "Self-Calibration of Unattended Ground Sensor Networks," *Proc. Advanced Sensor Consortium, ARL Federated Laboratory 5th Annual Symposium*, March 2001, pp. 63-70.

[22] S. Natkunanatham, A. Sipos, S. Avery, G. J. Pottie, and W. J. Kaiser, "A Signal Search Engine for Wireless Integrated Network Sensors," *Proc. Advanced Sensor Consortium, ARL Federated Laboratory 4th Annual Symposium*, March 2000, pp. 89-94.

[23] M. Chu, H. Haussecker, F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," to be published in the *International Journal on High Performance Computing Applications*, Special Issue on Sensor Networks, 2002.

[24] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd Edition, Academic Press.

[25] A. M. Sayeed, "Data-Driven Time-Frequency and Time-Scale Detectors," Proc. SPIE's 42nd Meeting, San Diego, July 1997.

[26] E. N. Onggosanusi, A. M. Sayeed, B. D. Van Veen, "Canononical Space-Time Processing for Wireless Communications," *IEEE Trans. Commun.*, pp. 1669-1680, Oct. 2000.

[27] A. J. Paulraj and C. B. Papadias, "Space-Time Processing for Wireless Communications," *IEEE Signal Processing Magazine*, pp. 49-83, Nov. 1997.

[28] C. M. Bishop, "*Neural networks for pattern recognition*", Clarendon press, Oxford, U.K., 1995.

Biographies:

Kerry D. Wong received the B.S. degree from Tsinghua University, Beijing, China, in June 2000. He is currently pursuing the M.S. degree in Electrical and Computer Engineering at the University of Wisconsin-Madison. His research interests are in signal processing and pattern recognition.

Dan Li received the BSEE degree from Tsinghua University, Beijing, China, in 1996, M.S. degree in Biomedical Engineering from University of Kentucky in 1999, and MSEE degree from University of Wisconsin-Madison in 2001. He is currently a Ph.D. student in Electrical and Computer Engineering at University of Wisconsin-Madison. His research interests are statistical signal processing and communication.

Yu Hen Hu is a faculty member at the Department of Electrical and Computer Engineering, University of Wisconsin, Madison. He received BSEE from National Taiwan University, and MSEE and PhD degrees from University of Southern California. Prior to joining University of Wisconsin, he was faculty in the Electrical Engineering Department of Southern Methodist University, Dallas, Texas. His research interests include multimedia signal processing, artificial neural networks, fast algorithms and design methodology for application specific micro-architectures, as well as computer aided design tools. He has published more than 180 technical papers in these areas.

Dr. Hu is a fellow of IEEE. He is a former associate editor (1988-1990) for the IEEE Transaction of Acoustic, Speech, and Signal Processing in the areas of system identification and fast algorithms. He is currently associate editor of Journal of VLSI Signal Processing. He is a founding member of the neural network signal processing technical committee of IEEE signal processing society and served as chair from 1993-1996. He is a former member of VLSI signal processing technical committee of the signal processing society. He served as the secretary of the IEEE signal processing society (1996-1998).

Akbar M. Sayeed received the B.S. degree from the University of Wisconsin-Madison in 1991, and the M.S. and Ph.D. degrees in 1993 an 1996, respectively, from the University of Illinois at Urbana-Champaign, all in electrical and computer engineering. While at the University of Illinois, he was a research assistant in the Coordinated Science Laboratory and was also the Schlumberger Fellow from 1992-1995. During 1996-1997, he was a postdoctoral fellow at Rice University. Since August 1997, he has been with the University of Wisconsin-Madison, where he is currently an Assistant Professor in Electrical and Computer Engineering. His research interests are in wireless communications, statistical signal processing, wavelets and time-frequency analysis.

Dr. Sayeed received the NSF CAREER Award in 1999 and the ONR Young Investigator Award in 2001. He is currently serving as an Associate Editor for the IEEE Signal Processing Letters.