

Real-Time Landmine Detection with Ground-Penetrating Radar Using Discriminative and Adaptive Hidden Markov Models

Hichem Frigui

*Department of Computer Engineering and Computer Science, University of Louisville, Louisville, KY 40292, USA
Email: h.frigui@louisville.edu*

K. C. Ho

*Department of Electrical and Computer Engineering, University of Missouri-Columbia, Columbia, MO 65211, USA
Email: hod@missouri.edu*

Paul Gader

*Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA
Email: pgader@cise.ufl.edu*

Received 25 October 2004; Revised 3 March 2005; Recommended for Publication by Fulvio Gini

We propose a real-time software system for landmine detection using ground-penetrating radar (GPR). The system includes an efficient and adaptive preprocessing component; a hidden Markov model- (HMM-) based detector; a corrective training component; and an incremental update of the background model. The preprocessing is based on frequency-domain processing and performs ground-level alignment and background removal. The HMM detector is an improvement of a previously proposed system (baseline). It includes additional pre- and postprocessing steps to improve the time efficiency and enable real-time application. The corrective training component is used to adjust the initial model parameters to minimize the number of misclassification sequences. This component could be used offline, or online through feedback to adapt an initial model to specific sites and environments. The background update component adjusts the parameters of the background model to adapt it to each lane during testing. The proposed software system is applied to data acquired from three outdoor test sites at different geographic locations, using a state-of-the-art array GPR prototype. The first collection was used as training, and the other two (contain data from more than 1200 m² of simulated dirt and gravel roads) for testing. Our results indicate that, on average, the corrective training can improve the performance by about 10% for each site. For individual lanes, the performance gain can reach 50%.

Keywords and phrases: landmine detection, hidden Markov models, corrective training, adaptive preprocessing.

1. INTRODUCTION

Detection and removal of landmines is a serious problem affecting civilians and soldiers worldwide. It is estimated that more than 100 million landmines are buried in more than 80 countries around the world, and that 26 000 people, mostly civilians, a year are either killed or maimed by a landmine [1, 2]. The detection problem is compounded by the large variety of landmine types, differing soil conditions, temperature and weather conditions, and varying terrain, to name a few. Detection and removal of landmines is therefore a significant problem, and has attracted several researchers in recent years. One challenge in landmine detection lies in plastic or low metal mines that cannot or are difficult to detect by traditional metal detectors.

Varieties of sensors have been proposed or are under investigation for landmine detection. The research problem for sensor data analysis is to determine how well signatures of landmines can be characterized and distinguished from other objects under the ground using returns from one or more sensors. Ground-penetrating radar (GPR) offers the promise of detecting landmines with little or no metal content. Unfortunately, landmine detection via GPR has been a difficult problem [3, 4, 5]. Although systems can achieve high detection rates, they have done so at the expense of high false-alarm rates. The key challenge to mine detection technology lies in achieving a high rate of mine detection while maintaining low level of false alarms. The performance of a mine detection system is therefore commonly measured by a receiver operating characteristics (ROC) curve

that jointly specifies rate of mine detection and level of false alarm.

Automated detection algorithms can generally be broken down into four phases: preprocessing, feature extraction, confidence assignment, and decision-making. Preprocessing algorithms perform tasks such as normalization of the data, corrections for variations in height and speed, removal of stationary effects due to the system response, and so forth. Methods that have been used to perform this task include wavelets and Kalman filters [6, 7], subspace methods and matching to polynomials [8], and subtracting optimally shifted and scaled reference vectors [9]. Feature extraction algorithms reduce the preprocessed raw data to form a lower-dimensional, salient set of measures that represent the data. Principal component (PC) transforms are a common tool to achieve this task [10, 11]. Other feature analysis approaches include wavelets [6], image processing methods of derivative feature extraction [12], curve analysis using Hough and Radon transforms [13], as well as model-based methods [14]. Confidence assignment algorithms can use methods such as Bayesian [13], hidden Markov models [12], fuzzy logic [15, 16], rules and order statistics [17], neural networks, or nearest-neighbor classifiers [11, 18], to assign a confidence that a mine is present at a point. Decision-making algorithms often postprocess the data to remove spurious responses and use a set of confidence values produced by the confidence assignment algorithm to make a final mine/no-mine decision.

In [12], hidden Markov modeling was proposed for detecting both metal and nonmetal mine types using data collected by a moving-vehicle-mounted GPR system. This (baseline) system uses both continuous and discrete HMMs, and has proved that HMM techniques are feasible and effective for landmine detection. In comparison with energy-based methods and fuzzy gradient-based detector, the HMM technique performed significantly better than the former, and achieved comparable performance to the latter [12]. The baseline discrete HMM was trained by conventional methods of vector quantization and the Baum-Welch algorithm. For the baseline continuous HMM, the parameters were initialized using clustering methods and learned using the Baum-Welch algorithm. The performance of the continuous HMM was slightly better than the discrete HMM, and the fusion of the two resulted in a better performance than those of the individual detectors.

In [19], minimum classification error (MCE) training was proposed to improve the performance of the discrete baseline HMM. After the initial training, this algorithm uses a sequential generalized probabilistic descent algorithm that minimizes an empirical loss function to estimate the mine/background model parameters. An evolutionary algorithm, based on fitness score of classification accuracy, was used to generate and select codebooks. The MCE-based discrete HMM achieved a significant performance improvement over the baseline system.

In this paper, we propose and evaluate a complete real-time software system for landmine detection using GPR. The detector is based on continuous HMM, and is an improved

version of the baseline system proposed in [12]. First, we propose a different preprocessing technique based on frequency-domain processing to perform ground-level alignment and background removal. This preprocessing approach is different from the one used in the baseline detector as the two systems were developed for different GPR prototypes, and the proposed system was designed to operate in a real-time mode. Second, pre- and postprocessing steps of the sequence observations were added to improve the time efficiency of the detector and enable real-time application. Third, an improved model training approach is proposed. The parameters of the baseline system were learned using the Baum-Welch algorithm. This standard approach does not guarantee minimization of the classification error rate, and cannot be used in an online mode to adapt the model parameters to different geographical sites and environments. The proposed system uses a heuristic corrective training procedure to adjust the initial parameters to minimize the number of misclassification sequences. Our approach is based on the optimal discriminative training (ODT) proposed by Mizuta and Nakajima [20]. This training could be used offline, using a signature library, to adjust the parameters of a generic model. It could also be used in a real-world operational mode, using feedback on which measurements are mines and false alarms once they are dug, to adapt the initial model to specific sites and environments. Fourth, we propose a dynamic background model that continuously adjusts its parameters to adapt it to the different lanes during testing. The proposed software system is applied to data acquired from several outdoor test sites, using a state-of-the-art array GPR prototype.

The rest of the paper is organized as follows. In Section 2, we provide related background on HMM, and different training methods. In Section 3, we give an overview of the GPR data. In Section 4, we describe the different components of the proposed system. In Section 5, the experimental data, training procedures, and results are discussed and summarized. Concluding remarks are given in Section 6.

2. BACKGROUND

2.1. Hidden Markov models

An HMM is a model of a doubly stochastic process that produces a sequence of random observation vectors at discrete times according to an underlying Markov chain. At each observation time, the Markov chain may be in one of N_s states $\{s_1, \dots, s_{N_s}\}$ and, given that the chain is in a certain state, there are probabilities of moving to other states. These probabilities are called the transition probabilities. An HMM is characterized by three sets of probability density functions, the transition probabilities (\mathbf{A}), the state probability density functions (\mathbf{B}), and the initial probabilities ($\boldsymbol{\pi}$).

Let T be the length of the observation sequence (i.e., number of time steps), let $\mathbf{O} = \{O_1, \dots, O_T\}$ be the observation sequence, and let $\mathbf{Q} = \{q_1, \dots, q_T\}$ be the state sequence. The compact notation

$$\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}) \quad (1)$$

is generally used to indicate the complete parameter set of the HMM model. In (1), $\mathbf{A} = [a_{ij}]$ is the state transition probability matrix, where $a_{ij} = \Pr(q_t = j \mid q_{t-1} = i)$ for $i, j = 1, \dots, N_s$; $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = \Pr(q_1 = s_i)$ are the initial state probabilities; and $\mathbf{B} = \{b_i(\mathbf{O}_t), i = 1, \dots, N\}$, where $b_i(\mathbf{O}_t) = \Pr(\mathbf{O}_t \mid q_t = i)$ is the set of observation probability distribution in state i .

An HMM is called continuous if the observation probability density functions are continuous and discrete if the observation probability density functions are discrete. In the case of the discrete HMM, the observation vectors are commonly vector quantized into a finite set of symbols, $\{v_1, v_2, \dots, v_M\}$, called the codebook. Each state is represented by a discrete probability density function and each symbol has a probability of occurring given that the system is in a given state. In other words, \mathbf{B} becomes a simple set of fixed probabilities for each class, that is, $b_i(\mathbf{O}_t) = b_i(k) = \Pr(v_k \mid q_t = i)$, where v_k is the symbol of the nearest code book of \mathbf{O}_t . In the continuous HMM, $b_i(\mathbf{O}_t)$'s are defined by a mixture of some parametric probability density functions. The most common parametric pdf used in continuous HMM is the mixture Gaussian density:

$$b_i(\mathbf{O}_t) = \sum_{m=1}^{M_i} c_{im} b_{im}(\mathbf{O}_t), \quad i = 1, \dots, N, \quad (2)$$

where M_i is the number of components in state i , c_{im} is the mixture coefficient for the m th mixture component in state i , and satisfies the constraints $c_{im} \geq 0$, and $\sum_{m=1}^{M_i} c_{im} = 1$, for $i = 1, \dots, N$, and $b_{im}(\mathbf{O}_t)$ is a K -dimensional multivariate Gaussian density with mean μ_{im} and covariance matrix \mathbf{C}_{im} .

Given the form of the hidden Markov model defined in (1), Rabiner [21] defines three key problems of interest that must be solved for the model to be useful in real-world applications: (i) the classification problem, (ii) the problem of finding an optimal state sequence, and (iii) the problem of estimating the model parameters. The classification problem involves computing the probability of an observation sequence $\{\mathbf{O} = \{\mathbf{O}_1, \dots, \mathbf{O}_T\}\}$ given a model λ , that is, $\Pr(\mathbf{O} \mid \lambda)$. Bayesian methods can be used to obtain the probability of the model given the observation. This probability can be computed with $O(TN^2)$ computations. In most applications, it often turns out that computing an optimal state sequence is more useful than $\Pr(\mathbf{O} \mid \lambda)$. There are several possible ways of finding an optimal state sequence associated with the given observation sequence, depending on the definition of the optimal state sequence, that is, there are several possible optimality criteria. One that is particularly useful is to maximize $\Pr(\mathbf{O}, \mathbf{Q} \mid \lambda)$ over all possible state sequences \mathbf{Q} . The Viterbi algorithm [22] is an efficient, formal technique for finding this maximum state sequence and associate probability. The third problem is the training problem: How does one estimate the parameters of the model? The problem is difficult because there are several levels of estimation required in an HMM. First of all, the states themselves must be estimated. Then the model parameters $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ need to be estimated.

In the discrete HMM, first the codebook is determined, usually using the K-means [21], or other vector quantization algorithms. Then the parameters $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ are estimated iteratively using the Baum-Welch algorithm [23]. In the continuous HMM, and for the case of Gaussian mixture density functions, the mixture component parameters, μ_{im} , \mathbf{C}_{im} , c_{im} , are first initialized (usually by clustering the training data), and then the continuous version of Baum-Welch is used to learn $(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$.

2.2. HMM training for minimizing classification error

The standard approach to estimate the HMM parameters is to use the expectation-maximization (EM) algorithm [24], also known as the forward-backward or Baum-Welch (BW) algorithm [23] in this context, to find the maximum-likelihood (ML) estimator. Unfortunately, ML training does not guarantee minimization of the classification error rate. Moreover, assumptions such as independent observations or state transition probabilities being dependent only on one previous state may not be valid, and may result in a sub-optimal performance. To alleviate these problems, several alternative training algorithms have been used mainly in the area of speech recognition. These methods can be roughly divided into three categories: maximum mutual information (MMI) training, minimum classification error (MCE) training, and corrective training. In MMI training, the HMM parameters are estimated by maximizing the mutual information between an observation sequence and the corresponding sequence of class labels. For the implementation of MMI, there is no efficient and robust procedure that is guaranteed to converge to an optimal solution. In [25, 26], the objective function is optimized using gradient search techniques with projection on constraints. In MCE training, the HMM parameters are estimated by formulating a loss function that incorporates the classification error rate over a set of training data [27, 28]. The empirical loss function is minimized using a generalized probabilistic descent algorithm.

Unlike MMI and MCE trainings, corrective training is motivated by intuition rather than theoretical optimization. It was first introduced by Bahl et al. [29] for the discrete HMM. A similar approach, called optimal discriminative training (ODT) was proposed by Mizuta and Nakajima [20] for the continuous HMM. In this approach, an initial model is used to classify the training samples, and the parameters are adjusted when an error is observed to avoid repeating the misclassification.

3. GPR DATA

3.1. GPR sensor

The GPR sensor system is mounted underneath a vehicle to provide a three-meter detection swath [30]. When the vehicle moves, the GPR transmit antenna sends out pulses that penetrate to the ground and the receive antenna captures the returned signal in order to facilitate the detection of landmine.

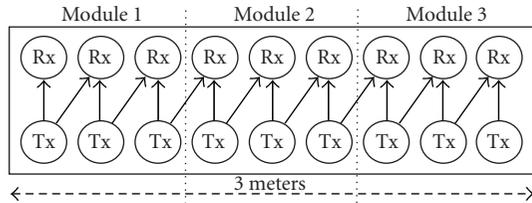
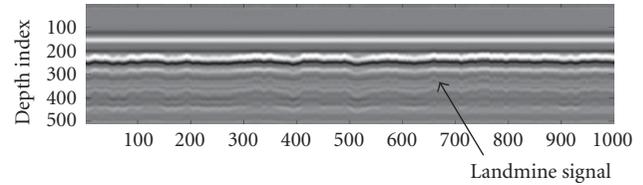


FIGURE 1: Arrangement of the 9 GPR radars.

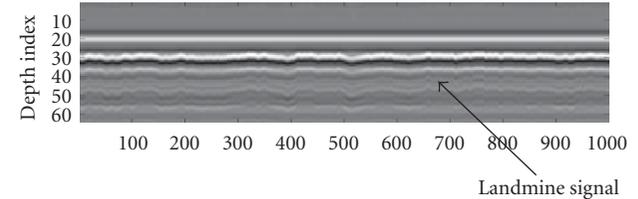
The GPR sensor is a time-domain, transient signal radar that gives target signal response information of a different nature than that from more conventional continuous-wave radars. The GPR has a pulse width of 1.258-nanosecond full width at half maximum (FWHM). Its operating frequency range is from 320 kHz to about 2 GHz, and its sampling rate is 136.6 GHz. The electromagnetic design of the impulse radiating antennas (IRAs) gives the very broadband radiation properties that allow for the required transient response. The impulse generator employed in the transmitter is designed to have a very fast rise and a slow decay in order to enhance both high-frequency and low-frequency components of the signals. The spectrum of the radar signal and further details of the hardware GPR system are documented in [31]. The receiver sampling system transforms the very fast real-time return signals from buried objects into much slower sampled-time data signals that are digitized, transmitted, and processed. In particular, at each location, a data vector of 2048 samples is acquired, which is then down-sampled to 512 points for storage and processing. The GPR acquires a vector sample in every 7.5 cm spatial distance. Each vector sample is then passed onto the software algorithms for the detection of landmine.

The GPR sensor system has 3 modules, and each module contains three transmit-receive radar pairs. Each module covers one-meter wide and the 3 modules together cover a three-meter swath (see Figure 1). Each transmit-receive radar pair consists of a pulse generator, a transmit antenna, a receive antenna, a low-noise preamplifier, and a sampling unit. A signal trigger generator and the delay block provides for the triggers for the complete sensor suite of all nine radars. The antenna design is based on that of the impulse radiating antenna as developed by the Air Force Research Laboratory, Albuquerque, New Mexico [32, 33, 34]. It consists of a parabolic conducting dish that is illuminated by two pairs of transmission-line fed arms from a drive point at the focal point of the paraboloid. The transmit and receive antennas have a diameter of 0.3 m.

Figure 1 shows the arrangement of the 9 transmit-receive radar pairs. In order to increase the spatial coverage in case a mine target is located between two adjacent radar pairs, each receiver receives not only the signal from its corresponding transmitter but also the signal from the left adjacent transmitter. This is accomplished by allowing each receiver antenna to operate with two samplers in each receiver operating simultaneously with its direct and left adjacent pulsers. We refer the reader to [31] for the details of this arrangement.



(a)



(b)

FIGURE 2: GPR data in a single channel: (a) data before downsampling in depth; (b) data after downsampling in depth by a factor of 8.

In real-time operational mode, the vehicle that carries the GPR system advances at a speed of 9 km/h, and the radar provides a three-meter detection swath. For instance, to scan and process 3×500 square meters, it takes about $0.5 \text{ km} / (9 \text{ km/h})$ or 3.33 minutes.

3.2. Data representation

The GPR data is arranged in a 3D array, with the cross-track, down-track, and depth corresponding to the three dimensions in the array. The total number of cross-track channels is 17 and each channel covers approximately 17 centimeters. The down-track sampling rate is 7.5 centimeters/sample. The GPR provides 512 depth data points in each sample location. Hence, for a 3-meter-times-500-meter lane, the GRR data has a size of $17 \times 6667 \times 512$. In order to reduce processing time, each data sample vector is downsampled from 512 to 64 by keeping one out of every eight data points. Figure 2(a) gives a segment of the original GPR data in a certain channel with 512 points in depth, and Figure 2(b) is the downsampled data by a factor of 8. Figure 3 shows typical A-scans with and without downsampling. Frequency-domain analysis indicates that the depth data points are oversampled. Moreover, extensive study has shown that downsampling in depth dimension by a factor of 8 does not introduce much degradation in the performance of the detector.

We should note here that the y -axis in Figure 2 and the x -axis in Figure 3 are related, but not equal, to the real depth. Computing the actual depth values would require the knowledge of the frequency bandwidth and the dielectric constant of the soil.

To facilitate real-time operation with parallel processing capability, the processing is performed independently from channel 1 to channel 17 at each down-track sample location. Processing begins with preprocessing of the data to remove

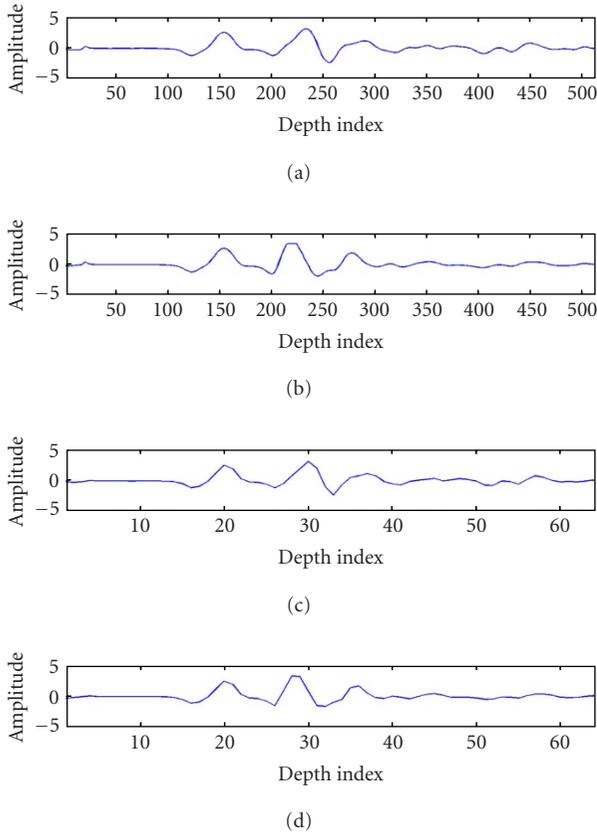


FIGURE 3: A-scans of GPR data before and after downsampling: (a) A-scan of background signal before downsampling, (b) A-scan of landmine and background signal before downsampling, (c) A-scan of background signal after 8:1 downsampling, and (d) A-scan of landmine and background signal after 8:1 downsampling.

system response, ground bounce, and noise. The HMM algorithm is then applied to generate an alarm confidence.

4. REAL-TIME LANDMINE DETECTION USING HMM

4.1. Data preprocessing

Preprocessing is an important step to enhance the mine signatures for detection. In general, preprocessing includes ground-level alignment and signal and noise background removal. In the proposed system, we assume that noise is much smaller than the background signal. In fact, the GPR hardware component performs waveform averaging to reduce background noise so that it is small enough to be ignored. The remaining background signals include the self-antenna coupling and ground reflection. Our preprocessing technique is designed to suppress this background signal.

Our preprocessing follows the shift-and-scale model proposed by Brunzell [9], where the current background vector sample is assumed to be a shifted and scaled version of a background reference. Background removal then requires the estimation of both shift and scale parameters, which are changing from sample to sample. The original shift-and-

scale model assumes the shift is an integer. In practice, this assumption may not be accurate as the shifting can be in the order of subpixels, and interpolation is needed to perform subpixel shifting. Unfortunately, interpolation increases the computation significantly, and prohibits real-time processing. In this paper, we propose an efficient subpixel shift and scale preprocessing in frequency domain so that subpixel shifting can be realized in a simple manner to reduce computation. The subpixel shift and the scale factor are obtained using the maximum-likelihood (ML) approach. Experimentally, we have found that subpixel shifting reduces the number of false alarms by at least a factor of 1.5 when compared to Brunzell's integer shifting.

Let \mathbf{x}_n denote the raw GPR data measured at a sample location, where n is the down-track location index. Since the processing is performed independently in each cross-track channel, the index for the channel has been dropped for notation simplicity. The vector \mathbf{x}_n contains the data points at different depth bins, that is, $\mathbf{x}_n = [x_n(1), x_n(2), \dots, x_n(M)]^T$, where $M = 64$ is the total number of depth bins in the down-sampled data as described in Section 3.2. The vector \mathbf{x}_n is modeled as

$$\mathbf{x}_n = \tilde{\mathbf{b}}_n + \mathbf{s}_n + \boldsymbol{\varepsilon}_n, \quad (3)$$

where $\tilde{\mathbf{b}}_n$ represents the background signal, \mathbf{s}_n is the response produced by a mine target or a clutter object, and $\boldsymbol{\varepsilon}_n$ is a vector of background noise. Note that the signal \mathbf{s}_n will be zero in the sample location that does not have mine or clutter object. The background return is relatively stable in normal environment and is considered to come from a global generic background vector $\mathbf{b} = [b(1), b(2), \dots, b(M)]^T$. Due to the surface roughness as the vehicle moves, $\tilde{\mathbf{b}}_n$ is modeled as a scaled and shifted version of \mathbf{b} , that is,

$$\tilde{\mathbf{b}}_n = a_n \mathbf{b}(\Delta_n), \quad (4)$$

where a_n is the scaling factor close to unity, Δ_n is the amount of shift in depth which is not necessarily an integer, and $\mathbf{b}(\Delta_n) = [b(1 + \Delta_n), b(2 + \Delta_n), \dots, b(M + \Delta_n)]^T$. The shift between two successive samples is typically very small, that is, $|\Delta_n - \Delta_{n-1}| \ll \Delta_n$, and less than an integer in most cases. This shift and scale background model has been used previously by Brunzell [9] for background removal, where the shift is assumed to be an integer. The problem is more challenging here since the shift is not necessarily an integer. The background return is much stronger than the signal \mathbf{s}_n and dominates the measurement. The objective here is to remove the background signal component so that the mine signature becomes apparent for detection. The background removal problem can be considered as finding the pair (a_n, Δ_n) given \mathbf{x}_n and \mathbf{b} , so that the background response can be subtracted out. In the absence of the signal \mathbf{s}_n and assuming Gaussian $\boldsymbol{\varepsilon}_n$, the maximum-likelihood solution is found by minimizing [9]

$$J'(\hat{a}_n, \hat{\Delta}_n) = (\mathbf{x}_n - \hat{a}_n \mathbf{b}(\hat{\Delta}_n))^T \mathbf{W}(\mathbf{x}_n - \hat{a}_n \mathbf{b}(\hat{\Delta}_n)), \quad (5)$$

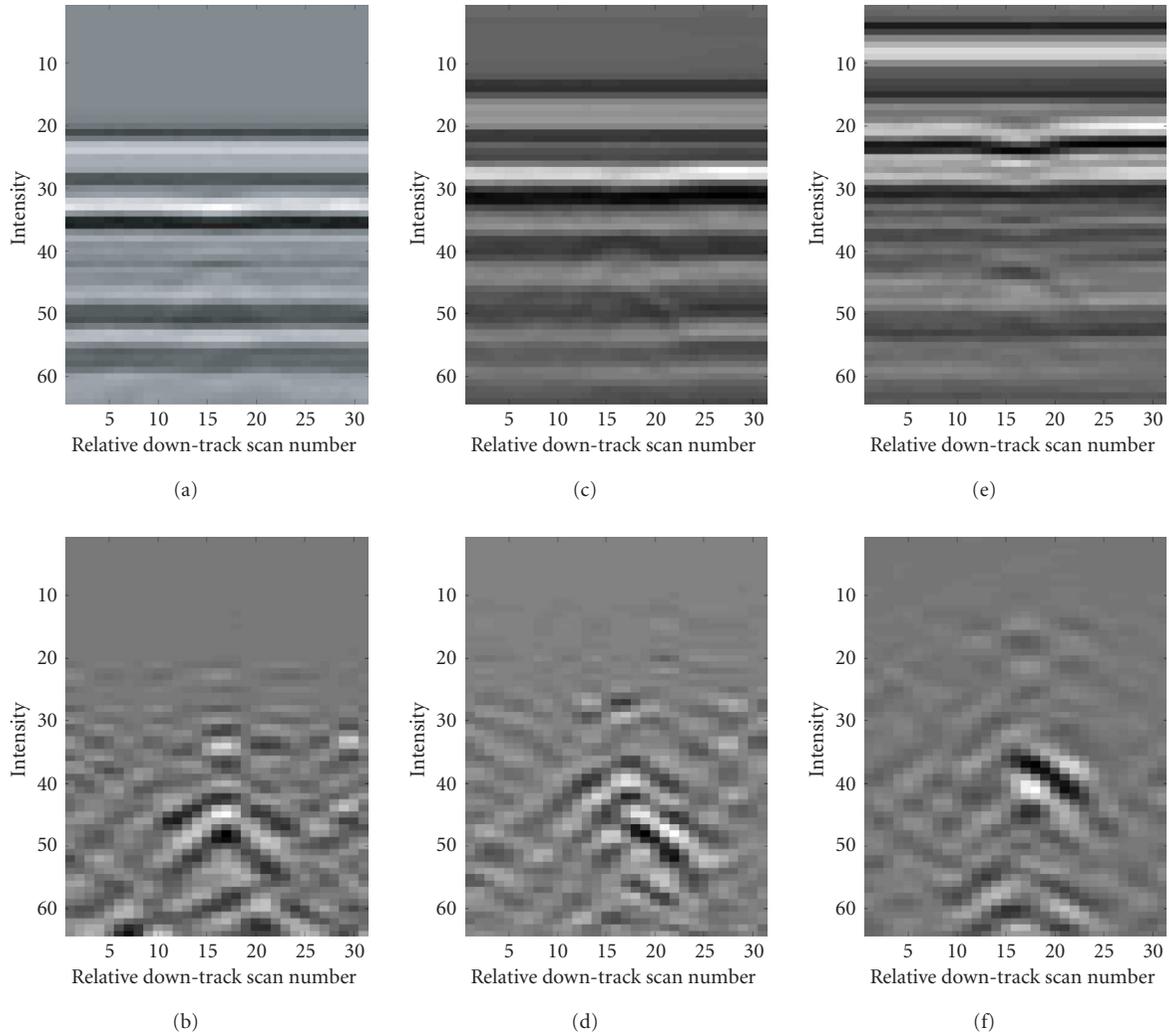


FIGURE 4: Preprocessed samples: (a) data from a plastic mine buried at 2'', (b) preprocessed result of (a), (c) data from a plastic mine buried at 3'', (d) preprocessed result of (d), (e) data from a metal mine buried at 6'', and (f) preprocessed result of (e).

where \mathbf{W} is the weighting matrix equal to $E[\boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^T]^{-1}$. Optimizing (5) with respect to \hat{a}_n yields

$$\hat{a}_n = \frac{\mathbf{b}(\Delta_n)^T \mathbf{W} \mathbf{x}_n}{\mathbf{b}(\Delta_n)^T \mathbf{W} \mathbf{b}(\Delta_n)}. \quad (6)$$

Substituting (6) into (5) gives

$$J'(\hat{\Delta}_n) = \mathbf{x}_n^T \mathbf{W} \mathbf{x}_n - \frac{(\mathbf{b}(\Delta_n)^T \mathbf{W} \mathbf{x}_n)^2}{\mathbf{b}(\Delta_n)^T \mathbf{W} \mathbf{b}(\Delta_n)}. \quad (7)$$

Because $\hat{\Delta}_n$ is not necessarily an integer, the elements of $\mathbf{b}(\hat{\Delta}_n)$ are formed by interpolation. That is,

$$b(i + \hat{\Delta}_n) = \sum_{m=1}^M b(m) \sin c(i + \hat{\Delta}_n - m), \quad i = 1, 2, \dots, M, \quad (8)$$

where $\sin c(*) = \sin(\pi*)/(\pi*)$. As a result, (7) is a nonlinear function with respect to $\hat{\Delta}_n$ and numerical search is necessary to find the solution $\hat{\Delta}_n$ that minimizes (7). Equation (8) requires M^2 multiplications to generate $\mathbf{b}(\hat{\Delta}_n)$ for a given $\hat{\Delta}_n$. Assuming the search space of $\hat{\Delta}_n$ has L values, then at a single sample location finding the optimum solution of $\hat{\Delta}_n$ requires an order of $M^2 L$ operations, which is computationally very intensive, and prohibits real-time processing. As an alternative, we propose to use a frequency-domain-based approach to reduce the computation.

The frequency-domain representation of (8) is

$$\mathbf{B}(\hat{\Delta}_n) = \text{diag} \{e^{j\omega_1 \hat{\Delta}_n}, e^{j\omega_2 \hat{\Delta}_n}, \dots, e^{j\omega_M \hat{\Delta}_n}\} \mathbf{B}, \quad (9)$$

where $\mathbf{B}(\hat{\Delta}_n)$ and \mathbf{B} are column vectors corresponding to the discrete Fourier transforms of $\mathbf{b}(\hat{\Delta}_n)$ and \mathbf{b} , and $\omega_m = 2\pi(m-1)/M$, $m = 1, 2, \dots, M$. Furthermore, \mathbf{W} in frequency

domain becomes the inverse of a diagonal matrix whose m th diagonal element is $P_\varepsilon(\omega_m)$, the power spectral density of ε_n at frequency ω_m . Hence, by representing (7) in the frequency domain and applying the Parseval theorem [18], the objective function to be maximized to obtain $\hat{\Delta}_n$ becomes

$$J(\hat{\Delta}_n) = \left| \sum_{m=1}^M \frac{1}{P_\varepsilon(\omega_m)} X_n(\omega_m) B^*(\hat{\Delta}_n, \omega_m) \right|^2, \quad (10)$$

where the constants that are independent of $\hat{\Delta}_n$ in (7) have been ignored, and the superscript $(*)$ represents complex conjugate. Note that $|B(\hat{\Delta}_n, \omega_m)|^2 = |B(\omega_m)|^2$ has been used so that the denominator in (7) is independent of $\hat{\Delta}_n$, and is therefore ignored. In (10), $X_n(\omega_m)$ and $B(\hat{\Delta}_n, \omega_m)$ represent the m th element of \mathbf{X}_n and $\mathbf{B}(\hat{\Delta}_n)$. Substituting (9) into (10) reduces to

$$J(\hat{\Delta}_n) = \left| \sum_{m=1}^M \left(\frac{1}{P_\varepsilon(\omega_m)} X_n(\omega_m) B^*(\omega_m) \right) e^{-j\omega_m \hat{\Delta}_n} \right|^2. \quad (11)$$

Note that the value inside the bracket is independent of $\hat{\Delta}_n$ and needs to be computed once. For a trial value $\hat{\Delta}_n$, (11) requires only $M + 1$ complex multiplications.

The computation can further be reduced by using the fact that $|\Delta_n - \Delta_{n-1}| \ll \Delta_n$. At the previous spatial instant $n - 1$, we obtained $\hat{\Delta}_{n-1}$. At the current instant n , we determine the relative translation $\hat{\delta}_n = \hat{\Delta}_n - \hat{\Delta}_{n-1}$ by maximizing

$$\tilde{J}(\hat{\delta}_n) = \left| \sum_{m=1}^M \left(\frac{1}{P_\varepsilon(\omega_m)} X_n(\omega_m) X_{n-1}^*(\omega_m) \right) e^{-j\omega_m \hat{\delta}_n} \right|. \quad (12)$$

The solution of $\hat{\Delta}_n$ is then equal to $\hat{\Delta}_{n-1} + \hat{\delta}_n$. The search space now reduces from L of $\hat{\Delta}_n$ to L_δ of $\hat{\delta}_n$, where $L_\delta < L$. Finding $\hat{\Delta}_n$ requires an order of ML_δ multiplications instead of M^2L in the time-domain approach.

Once $\hat{\Delta}_n$ has been determined, the final step is to perform background subtraction. Instead of shifting \mathbf{b} by $\hat{\Delta}_n$, we shift \mathbf{x}_n by $-\hat{\Delta}_n$ so that all the data samples will be aligned to a single ground reference point. Denote the aligned data sample \mathbf{x}_n as $\tilde{\mathbf{x}}_n$, which can be found by the inverse Fourier transform of $\text{diag}\{e^{j\omega_1 \hat{\Delta}_n}, e^{j\omega_2 \hat{\Delta}_n}, \dots, e^{j\omega_M \hat{\Delta}_n}\}^* \mathbf{X}_n$. From (3), $\tilde{\mathbf{x}}_n$ can be expressed as

$$\tilde{\mathbf{x}}_n = a_n \mathbf{b} + \mathbf{s}_n(-\hat{\Delta}_n) + \varepsilon_n(-\hat{\Delta}_n). \quad (13)$$

When using the previous aligned sample $\tilde{\mathbf{x}}_{n-1}$ as the background reference of \mathbf{x}_n , the background removed sample is

$$\begin{aligned} \mathbf{y}_n &= \tilde{\mathbf{x}}_n - \hat{a}_n \tilde{\mathbf{x}}_{n-1} \\ &= \mathbf{s}_n(-\hat{\Delta}_n) - \hat{a}_n \mathbf{s}_{n-1}(-\hat{\Delta}_{n-1}) \\ &\quad + \varepsilon_n(-\hat{\Delta}_n) - \hat{a}_n \varepsilon_{n-1}(-\hat{\Delta}_{n-1}), \end{aligned} \quad (14)$$

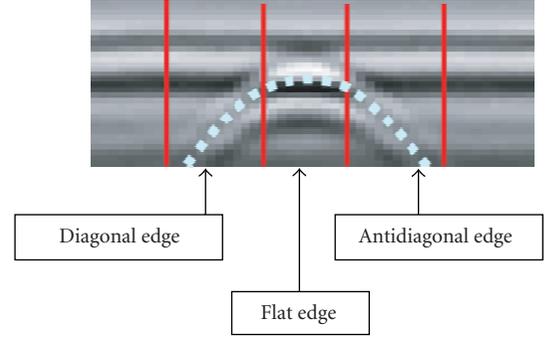


FIGURE 5: Features of a mine signature.

where \hat{a}_n is computed from (6). Alternatively, it can also be computed using $\hat{a}_n = (\tilde{\mathbf{x}}_n^T \mathbf{W} \tilde{\mathbf{x}}_{n-1}) / (\tilde{\mathbf{x}}_{n-1}^T \mathbf{W} \tilde{\mathbf{x}}_{n-1})$ by minimizing $\mathbf{y}_n^T \mathbf{y}_n$. The advantage of this solution is that we do not need to determine explicitly the background reference \mathbf{b} and its shifted version $\mathbf{b}(\hat{\Delta}_n)$.

The first two terms on the right-hand side of (14) represent the mine signal and the last two are noise. Note that the background removal results in a first-order difference of the signal component. Fortunately, the first difference is required in the HMM and hence will not introduce undesirable effect. Finally, a lowpass filter that has coefficients $[1, 3, 3, 1]$ is applied to every depth bin of the processed data \mathbf{y}_n along down-track to reduce noise.

The following example shows some preprocessing examples extracted from data that were collected at a test site in the United States. The extracted regions correspond to signatures of antitank mines with diameters between 20 and 30 cm. Figures 4(a) and 4(b) show the data of a plastic mine buried at 2 inch deep before and after preprocessing. Figures 4(c) and 4(d) are another example of a plastic mine buried at 3 inch deep. Figures 4(e) and 4(f) are an example of a metal mine buried at 6 inch deep. As can be seen, the preprocessing is very effective in removing the ground response and enhancing the mine signature.

The above preprocessing algorithm is designed to suppress the background signal. However, it cannot discriminate between the reflection of mines and different clutter objects. In fact, the reflection from most clutter objects could generate high energy return, and would be the main source of false alarms. The proposed feature extraction and HMM classifier, presented in the next section, are designed to discriminate between mines and clutter objects and to reduce the number of false alarms.

4.2. Feature extraction

Landmines (and other buried objects) appear in time-domain GPR as shapes that are similar to hyperbolas corrupted by clutter. Thus, the feature representation adopted by the HMM-based system is based on the degree to which edges occur in the diagonal and antidiagonal directions, and the features were extracted to accentuate these edges.

Figure 5 displays a hyperbolic curve superimposed on a pre-processed metal mine signature to illustrate the features of a typical mine signature.

$$\begin{aligned} D_y(x, y, z) &= \frac{[S(x, y + 2, z) + 2S(x, y + 1, z) - 2S(x, y - 1, z) - S(x, y - 2, z)]}{3}, \\ D_{yy}(x, y, z) &= \frac{[D_y(x, y + 2, z) + 2D_y(x, y + 1, z) - 2D_y(x, y - 1, z) - D_y(x, y - 2, z)]}{3}. \end{aligned} \quad (15)$$

The derivative values are then normalized along the y direction using

$$N(x, y, z) = \frac{D_{yy}(x, y, z) - \mu(x, z)}{\sigma(x, z)}, \quad (16)$$

where $\mu(x, z)$ and $\sigma(x, z)$ are the running mean and standard deviation updated using a buffer of scans available during system operation.

The down-track dimension is taken as the time variable in the HMM model. The goal is to produce a confidence that a mine is present at various positions, (x, y) , on the surface being traversed. To fit into the HMM context, a sequence of observation vectors must be produced for each point. These observation vectors encode the degree to which edges occur in the diagonal and antidiagonal directions.

The observation vector at a point (x_s, y_s) consists of a set of 15 features that are computed on a normalized array of GPR data of size 32×8 . Let x_s and y_s be given and let \mathbf{A} denote the array

$$\mathbf{A} = \mathbf{A}(y, z) = N(x, y, z), \quad (17)$$

where $x = x_s$, $y = \{y_s - 3, \dots, y_s + 4\}$, and $z = \{1, 2, \dots, 32\}$. The array \mathbf{A} is then broken into positive and negative parts according to the formulas

$$\begin{aligned} A^+(y, z) &= \begin{cases} A(y, z) & \text{if } A(y, z) > 1, \\ 0 & \text{otherwise,} \end{cases} \\ A^-(y, z) &= \begin{cases} -A(y, z) & \text{if } A(y, z) < -1, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (18)$$

Next, for each point in the positive and negative parts of \mathbf{A} , the strengths of the diagonal and antidiagonal edges are estimated and are used to define the 15-dimensional observation vector associated with the point (x_s, y_s) . We refer the reader to [12] for a more detailed description of the observation vector.

4.3. Improved continuous HMM-based classifier for landmine detection

The proposed HMM classifier consists of two HMM models, one for mine and one for background. The mine model,

Let $S(x, y, z)$ denote the preprocessed three-dimensional GPR data as illustrated in Figure 6. The down-track second derivative is first estimated on the raw data using

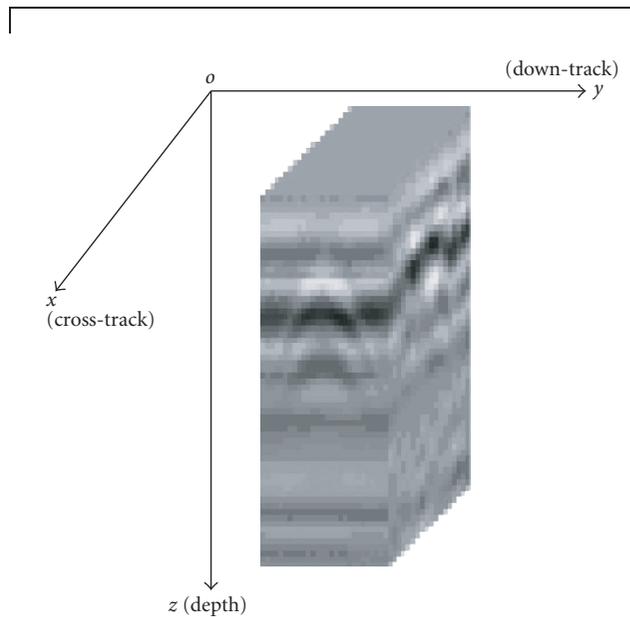


FIGURE 6: A collection of few GPR scans.

λ^m , is designed to capture the hyperbolic spatial distribution of the features. λ^m has 3 states which correspond to the rising edge, flat, and decreasing edge. Each state is represented by 3 Gaussian components. The mine model is illustrated in Figure 7. The background model is needed to capture the background characteristics and to reject false alarms. Each of the 17 channels is treated independently from the others, and has its own background model, λ^{b_c} . In addition to allowing each channel to have a model that reflects its own data, this decoupling allows the channels to be processed in parallel, and thus facilitating real-time operation. All λ^{b_c} (for $c = 1, \dots, 17$) have 3 states and 3 Gaussian components per state. The model architecture is illustrated in Figure 8.

Each HMM produces a probability by backtracking through model states using the Viterbi algorithm [22]. The probability value produced by λ^m (λ^{b_c}) can be thought of as an estimate of the probability of the observation sequence given that there is a mine (background) present.

The proposed HMM classifier is based on a previous work by Gader et al. [12]. In the following, we only outline the modifications we made to the baseline system to improve its performance and efficiency.

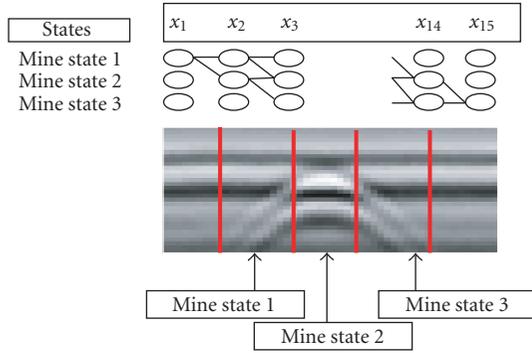


FIGURE 7: Illustration of the HMM mine model.

4.3.1. Observation sequence preprocessing

Most mines have signatures of different sizes, and their distribution cannot be captured efficiently by a fixed number of observations. To address this problem, the baseline system uses a mine model with 5 states. The first and last states are background states. During testing, each sequence is allowed to start in either the background state or the first mine state. Moreover, two optimal state sequences are computed: one assuming the model assigns the last observation to the third mine state, and the other assuming the last ob-

servation is assigned to the background state. The state sequence with the highest probability is taken as the model output.

The proposed software system uses observation sequences of variable length. This approach is more efficient as only 3 states are needed for the mine model, and only one optimal state sequence is needed. Initially, each sequence candidate has a fixed number of observations, T_{\max} . Before feeding the sequence to the HMM classifier, we ignore all leading and trailing observations with weak feature vectors. In other words, we exclude observation O_t from the sequence if $\|O_t\| < \theta_{\min}$. This process results in a sequence of length T , where $T < T_{\max}$. If the resulting sequence does not include enough observations (i.e., $T < T_{\min}$), then we classify it as background without any further processing. In all the results reported in this paper, we use $\theta_{\min} = 0.1$, and $T_{\min} = 7$. The final probability of each sequence is normalized by the sequence length to avoid any biases.

4.3.2. Confidence value assignment

The confidence value assigned to each observation sequence, $\text{Conf}(\mathbf{O})$, depends on (1) the probability assigned by the mine model, $P(\mathbf{O}|\lambda^m)$; (2) the probability assigned by the background model, $P(\mathbf{O}|\lambda^{b_c})$; and (3) the optimal state sequence. In particular, we use

$$\text{Conf}(\mathbf{O}) = \begin{cases} 0 & \text{if } \frac{\log(P(\mathbf{O}|\lambda^m))}{T} < \text{Min_Prob}, \\ 0 & \text{if } s_{1,2} \neq 1 \text{ or } s_{T,T-1} \neq 3, \\ \log(P(\mathbf{O}|\lambda^m)) - \frac{\log(P(\mathbf{O}|\lambda^{b_c}))}{T} & \text{otherwise.} \end{cases} \quad (19)$$

The mine and background probabilities are normalized by the sequence length to avoid any bias that may be introduced by using a different number of observations. The first condition requires that the probability assigned by the mine model exceed a threshold for the confidence to be nonzero. This condition prevents observation sequences with low mine probability and low background probability (i.e., unfamiliar sequences not represented in the training data) from having a positive confidence value. We set the threshold Min_Prob to -5 . The second condition requires that at least the first two states in the optimal state sequence correspond to the rising edge of the signature (i.e., mine state 1), and at least the last two states correspond to the falling edge (i.e., mine state 3). This condition is helpful in rejecting several false alarms that do not follow the general hyperbolic shape of the mine signatures, but can have a high confidence value (due to very low background probability). We should note here that in (19), if one of the first two conditions is true, there is no need to evaluate the probability of the observation using the background model.

4.4. Corrective training for mine/background HMM models

As mentioned earlier, the BW algorithm which is the standard approach to estimate the HMM parameters does not guarantee minimization of the classification error rate. Moreover, in the case of GPR data, the assumption that the observations are independent is not true. To overcome these difficulties, our system uses a combination of BW and corrective training to learn one global mine model and a background model for each channel that minimize the classification error rate on the training data.

Corrective training is appealing because it is not sensitive to the assumed model type, does not involve complex optimization, and relies on the training data to minimize the number of misclassified samples. First, we use the BW procedure to estimate the parameters. Then, we use the Viterbi algorithm to compute the optimal state sequence and its probability for each training signature for both the mine and background models. Then, whenever a misclassification is

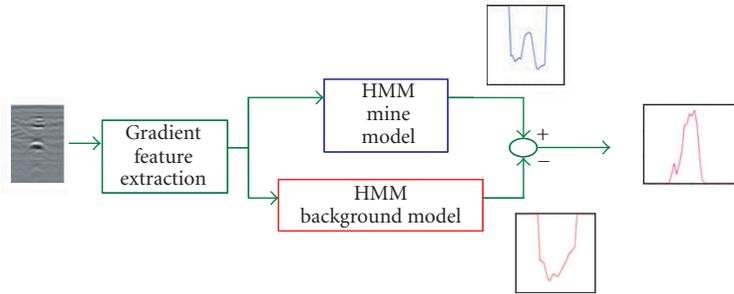


FIGURE 8: Illustration of the HMM-based model architecture.

```

Read mine and background signature libraries
Estimate initial mine model parameters using BW procedure
Estimate initial background model parameters using BW procedure
Repeat
  For each signature observation sequence  $\mathbf{O}$ 
    Let  $c$  denote the channel from which signature is extracted
    Use Viterbi procedure to compute  $P(\mathbf{O}|\lambda^m)$  &  $P(\mathbf{O}|\lambda^{b_c})$ 
    If  $(P(\mathbf{O}|\lambda^m) \geq P(\mathbf{O}|\lambda^{b_c}))$  classify  $\mathbf{O}$  as mine
    Else Classify  $\mathbf{O}$  as background
    If signature is misclassified
      Let  $r$  denote the correct model
      Let  $w$  denote the incorrect model
      For each observation  $O_t$  of the sequence  $\mathbf{O}$ 
         $i =$  state assigned to  $O_t$  by model  $r$ 
         $k =$  most probable component of state  $i$  of model  $r$ 
        
$$k = \arg \max_{m=1, \dots, M_i} b_m^r(O_t)$$

        
$$\mu_{ki}^r(\text{new}) = \mu_{ki}^r(\text{old}) + \eta(x(t) - \mu_{ki}^r(\text{old}))$$

         $j =$  state assigned to  $O_t$  by model  $w$ 
         $l =$  most probable component of state  $j$  of model  $w$ 
        
$$l = \arg \max_{m=1, \dots, M_j} b_m^w(O_t)$$

        
$$\mu_{lj}^w(\text{new}) = \mu_{lj}^w(\text{old}) - \eta(x(t) - \mu_{lj}^w(\text{old}))$$

      End for loop
    End if loop
  End for loop
  Reestimate mine model parameters using BW procedure
  Reestimate background models parameters using BW procedure
Until (convergence or maximum no. of passes reached)

```

ALGORITHM 1: Corrective training for mine/background HMM models.

observed, the parameters of λ^m and λ^{b_c} are adjusted to reduce the likelihood of repeating this error. In particular, the Gaussian means of the most probable component of the states in the Viterbi paths are adjusted using an LVQ-type [35] learning rule. The above steps are repeated until all the training data is classified correctly, or the maximum number of passes is reached. The proposed corrective training algorithm is outlined in Algorithm 1.

In this training algorithm, (20) is an LVQ-type [35] learning rule. The mean vectors of the Gaussian components are moved closer to the observations of the correctly classified sequences, and further from the observations of the mis-

classified sequences. In (20), the constant η is the learning rate, and has similar behavior as in standard LVQ algorithms. If η is too small, then several passes would be needed for the algorithm to converge. On the other hand, if η is too large, the training algorithm may not converge as large corrections made by some sequences may be reversed by other sequences. In this paper, we report the results using $\eta = 1.0e-3$ for the mine model, and $\eta = 1.0e-4$ for the background model. The reason for using a smaller rate for the background is that this model is, in general, less consistent than the mine model, and the magnitude of the error (second part of (20)) is usually larger than that caused by the mine model.

```

Let  $\lambda^{bc}$  be the background model of channel  $c$  estimated using training data,
 $\mu_{ij}^c$  = mean of component  $i$  of state  $j$  of  $\lambda^{bc}$ 
 $C_{ij}^c$  = covariance matrix of component  $i$  of state  $j$  of  $\lambda^{bc}$ 
 $n_{ij}$  = number of training observations that contributed to the estimation of  $\mu_{ij}^c$  and  $C_{ij}^c$ .
Let  $N_{ij} = K \times n_{ij}$ ;  $S_{ij} = N_{ij} \times \mu_{ij}^c$ ;  $SSq_{ij} = N_{ij} \times [C_{ij}^c + \mu_{ij}^c(\mu_{ij}^c)^T]$  denote the
sufficient statistics of Gaussian component  $i$  of state  $j$ .
For each tested observation sequence  $\mathbf{O}$  with  $(\text{Conf}(\mathbf{O}) < \text{Min\_Conf})$ 
  For each observation  $O_t$  of  $\mathbf{O}$ 
     $j$  = state assigned to  $O_t$ 
     $i$  = most probable component of state  $j$  :  $i = \arg \max_{m=1, \dots, M_j} b_{mi}(O_t)$ 
     $S_{ij} = S_{ij} + x_t$ 
     $SSq_{ij} = SSq_{ij} + x_t x_t^T$ 
     $N_{ij} = N_{ij} + 1$ 
  End
End
 $\mu_{ij}^c = \text{Sum}_{ij}/N_{ij}$ ;  $C_{ij}^c = \text{Sum}Sq_{ij}/N_{ij} - \mu_{ij}^c(\mu_{ij}^c)^T$ ; for  $i = 1, \dots, N$  and  $j = 1, \dots, M_j$ 

```

ALGORITHM 2: Incremental update of the background models.

As in most learning algorithms, the number of training iterations for the proposed corrective training algorithm is important. Generally, this number would depend on the learning rate, η , the dielectric constant of the soil from which the training signatures were extracted, the consistency of the training signatures, and the level of variations between the signatures used for the initial training and the signatures used for the subsequent corrective training. If the number of corrective training iterations is not sufficient, the parameters of the models would not adapt sufficiently to the given site. On the other hand, if too many corrective iterations are applied, the models may get overfitted. The standard approaches to address this issue is to either stop the training when the performance does not improve significantly, or use a cross validation data set and stop the training when the performance on the validation set (not used for training) starts to decrease. Currently, our system uses the former approach as our labeled training data is not large enough to be split into training and cross validation sets. Experimentally, we have found that only 5 to 10 iterations are needed when the corrective training algorithm is applied offline to the training signatures. Applying the algorithm for more iterations does not improve the performance of the detector any further.

The corrective training procedure described above operates on a signature library, and is for the offline mode. It can be easily modified to operate in a real-time mode and adapt the HMMs to different sites and environments using feedback on which measurements are mines and false alarms once they are dug. In this case, instead of reading the signatures, the algorithm starts by reading the lane ground truth. Then, the lane data is progressively read and processed as the GPR mounted vehicle mover over the lane. As in the offline mode, the parameters are adjusted whenever a misclassification occurs. To perform more than one iteration of corrective training, one could either drive over the lane multiple times, or simply store the data and reprocess it few more times when the end of the lane is reached.

In this real-time mode, the model parameters are adopted to new data that was not included in the initial training phase. Thus, it is expected that more iterations would be

needed. Experimentally, we have found that, for all the calibration lanes, 10 to 20 corrective training iterations are sufficient. Applying the algorithm for more iterations does not improve the performance any further.

4.5. Adaptive background model

The mine model is well defined and needs to be designed to capture the hyperbolic mine signatures. The background model, on the other hand, is not well defined. It is needed to model the background characteristics so as to reduce the number of false alarms, however, it is almost impossible to specify the characteristics of this model. This is because the background model is affected by several factors such as differing soil conditions, temperature and weather conditions, and varying terrain. In fact, the background can change between the start and the end of the same test lane as the hardware settings and weather conditions can change. Since it is not possible to collect training data that covers all different settings, our systems uses an adaptive background model. Our adaptive approach involves two main steps.

4.5.1. Adaptive data normalization

To maintain the same dynamic range, and use constant thresholds, we adaptively normalize the preprocessed data so that its values are normally distributed with zero mean and unit variance. The data statistics are dynamically updated using a buffer of the most recent observations. Observation sequences are added to the buffer only if their likelihood of belonging to the background model is high. We fix the buffer size to 50 observations, and as the buffer gets full, new observations would replace the old ones.

4.5.2. Incremental update of the background models

An initial background model, λ^{bc} , is estimated for each channel, c , by applying the corrective training procedure on the training signatures. While testing, observation sequences with high $P(\mathbf{O}|\lambda^{bc})$ and low $P(\mathbf{O}|\lambda^m)$ are used to adjust the parameters of the background model of channel c . In particular, we use Algorithm 2.

TABLE 1: Summary of the three data collections used in the experiments.

Collection	Location	Content			
(1) Signatures	Site 1	2 945 mine observation sequences.			
		5 737 false alarm/background observation sequences.			
(2) Lane data	Site 2	Lane 4	100 m × 3 m	14 mines	21 passes
		Lane 6	100 m × 3 m	10 mines	24 passes
		Lane 8	100 m × 3 m	19 mines	20 passes
		Lane 9	100 m × 3 m	18 mines	18 passes
(3) Lane data	Site 3	Lane 51	200 m × 3 m	21 mines	6 passes
		Lane 52	200 m × 3 m	16 mines	6 passes
		Lane 56	200 m × 3 m	21 mines	6 passes

5. EXPERIMENTS

5.1. Experimental data

GPR data collected from three different sites in the United States was used in our experiments. We will refer to these as collection 1, collection 2, and collection 3, respectively. Collection 1 contains mines and false alarms/background signatures extracted from data collected at site 1. These signatures were selected using a combination of ground truth and visual examination, and consist of 2 945 mines and 5 737 false alarms/background observation sequences. A more detailed description of this signature library and the extraction process can be found in [12]. This collection is used to learn the initial mine and background HMM model parameters using the basic Baum-Welch algorithm.

Collections 2 and 3 contain GPR lane data collected from sites 2 and 3, respectively. Some of these lanes were used for corrective training, and the remaining for testing. Testing and adapting the model parameters on lanes is the most representative of real-world operational mode. Site 2 has 4 lanes labeled lane 4, 6, 8, and 9. Site 3 has 3 lanes labeled lane 51, 52, and 56. Each lane is 3 m wide, and its length varies from 50 to 200 m. The ground truth, that is, location of the mines, of these calibration lanes is available, and we use it only during the corrective training or in the evaluation of the algorithm, and not in the HMM testing algorithm itself. Multiple data files were collected from each lane. Each file, referred to as one pass, corresponds to running a vehicle-mounted GPR system over the lane. A total of 83 passes were collected from site 2, and only one of these passes (from lane 4) was used for corrective training to adapt the HMMs to this site. Collection 3 includes 18 passes, and one of the passes (from lane 51) was used for training. Table 1 summarizes the data collections used in our experiments.

We should note here that although the lanes used in our experiments do not have explicitly emplaced clutter, they were set up to represent real on-road conditions that a vehicle-mounted mine detection system will encounter. Thus, they do include clutter objects such as void, rocks, and probably some metal pieces.

5.2. Basic HMM training

The signature library (i.e., collection 1) was used to train the basic HMM models. Using this collection, BW algorithm was used to estimate the HMM parameters $\Lambda = (\lambda^m, \lambda^b)$, where λ^m is estimated from the mine signatures and λ^b is estimated from the false alarms/background signatures. We will refer to these as the baseline models, or simply Λ_{base} .

5.3. Corrective HMM training using the signature library

Using Λ_{base} as initial model parameters, we applied the corrective training procedure to the signature library and adjusted the parameters to reduce the number of misclassified signatures. A total of 10 iterations were needed for the parameters to stabilize. We will refer to these models as Λ_{Sig} .

5.4. Corrective HMM training using lane data

The rationale of the proposed training approach is to adapt the model parameters to different geographical sites that have different characteristics due to variations in soil type and other conditions. Thus, we have selected one pass from collection 2 and used it (along with its ground truth) to adapt the baseline HMMs to this site. The training pass was selected from lane 8 as this lane contains more mines with more variety. In all experiments reported in this paper, a total of 20 corrective training iterations were performed, and intermediate model parameters were saved after each iteration. We will refer to these models as $\Lambda_{\text{S2-L8}}(k)$ (i.e., models adapted to site 2 using lane 8 after k iterations).

Similarly, we have adapted the baseline HMMs to site 3 by applying the corrective training using one pass from this collection (from lane 51). For this site, we obtained the intermediate models $\Lambda_{\text{S3-L51}}(k)$.

5.5. Experimental results

In this section, we present, analyze, and compare the results obtained using the corrective training procedure with those obtained using the baseline model. Our results could not be compared with the basic system introduced in [12], as these systems use different GPR prototypes, and thus, have pre-processed the data differently. However, we should point out

that the two systems are very similar. The major differences are the additional pre- and postprocessing steps outlined in Section 4.3. These steps do not affect the detection results significantly. They are added mainly to reduce the execution time, and thus, enable real-time implementation and testing. This is because (1) the sequences are no longer required to have a constant length of 15 (maximum length); (2) only 3 not 5 states are needed for the mine model; and (3) several sequences are rejected without testing them with the background models using the Viterbi algorithm.

The performance of the different model parameters was scored in terms of probability of detection (PD) versus false alarm rate (FAR). Confidence values were thresholded at different levels to produce receiver operating characteristics (ROC) curves. For a given threshold, a mine is detected if there is an alarm within 0.25 meters from the edge of the mine with confidence value above the threshold. Given a threshold, the PD for a lane or a set of lanes is defined to be the number of mines detected divided by the total number of mines. The FAR is defined as the number of false alarms per square meter.

Figure 9a shows the overall ROCs for all of the 83 passes of collection 2. The ROCs are displayed for the confidence values generated using the baseline model, Λ_{base} (no corrective training), the Λ_{sig} model (corrective training using signature library), and the $\Lambda_{\text{S2-L8}}(k)$ models (k iterations of corrective training using one pass from collection 2). As can be seen, when compared to the Λ_{base} ROC, the Λ_{sig} and $\Lambda_{\text{S2-L8}}(k)$ ROCs are shifted left (i.e., lower FAR for the same PD) and shifted up (higher PD for the same FAR). Thus, one can conclude that corrective training improves the overall detection rate.

To quantify the improvement in the detection rate, we compute the area under the ROC in the $[0, 0.02]$ FAR range.¹ For the rest of this paper, we will use the term *performance* to refer to this area. A perfect detector (detects all the mines with no false alarms) would have an area of 0.02. Figure 9b displays the values of these areas for the ROCs of the different models. As can be seen, the area jumps from 0.013 (for the baseline parameters) to 0.0134 for the Λ_{sig} parameters, and to 0.014 $\Lambda_{\text{S2-L8}}(20)$, achieving an overall performance gain of about 10%. As can be seen, the performance curve starts to flatten after 15 iterations. Performing the corrective training for more than 20 iterations does not improve the performance any further.

Figure 9 displays the results averaged over 83 passes. The performance (measured by the area under the ROC) for the individual passes varies significantly. For instance, for some passes, the performance remains constant. These passes are usually either “clean,” with all mines easily detectable, or very noisy with several weak mine signatures. Thus, the performance is either close to optimal, and cannot be improved any further, or poor and cannot be improved, as many of the mines could not be detected using

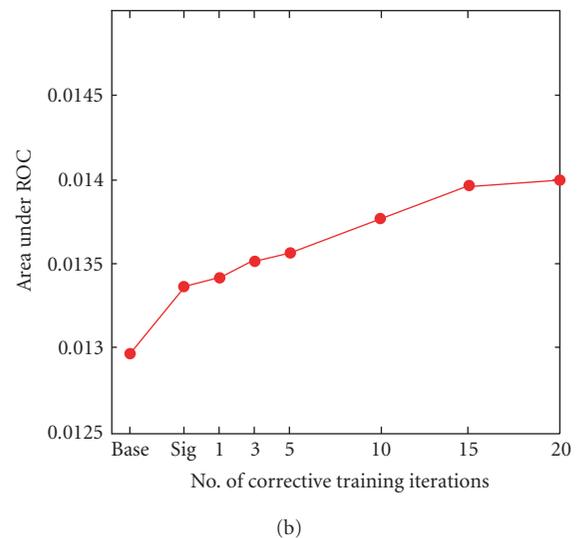
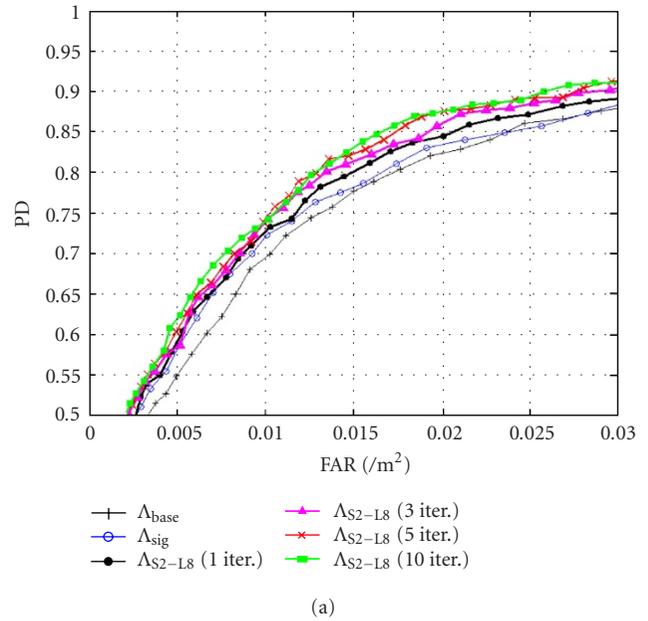


FIGURE 9: Average performance of the HMM detector with different model parameters on the 83 passes of collection 2. (a) ROCs with and without corrective training. (b) Improvement in the performance (area under ROC in the $[0, 0.02]$ FAR range) versus number of corrective training iterations.

the current preprocessing and feature representation techniques.

For most passes, the performance improves. Figures 10 and 11 display the results for two different passes over lanes 4 and 6, respectively, where the performance improves by about 50%. For Figure 10, the improvement occurs after the initial corrective training with the signature library, and more training with the lane data does not affect the results. On the other hand, for Figure 11, no significant improvement was obtained after the corrective training with the signature library. The largest performance gain was obtained using $\Lambda_{\text{S2-L8}}(10)$.

¹The performance criterion for the vehicle-mounted mine detection system technology was set to 0.02/ m^2 FAR at 90% PD, and is described in [36].

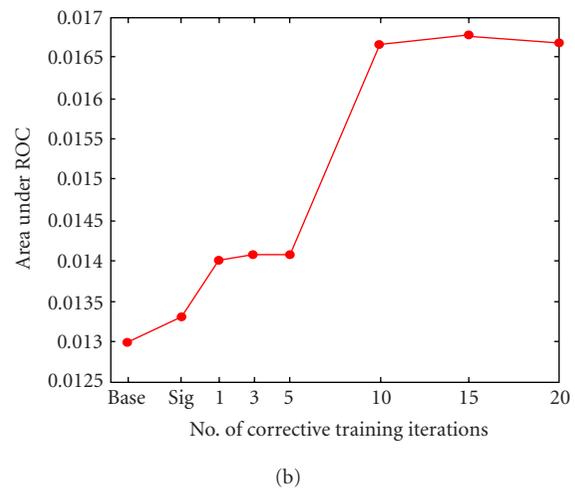
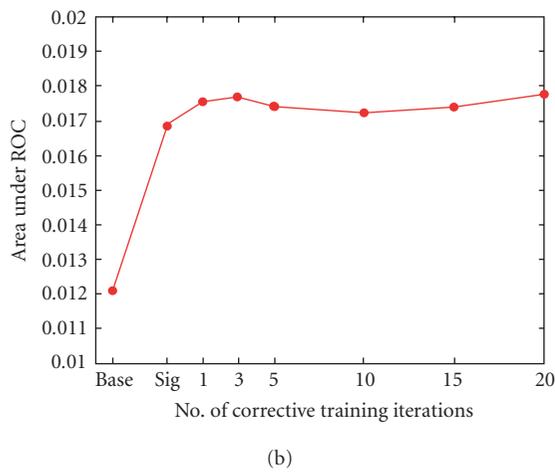
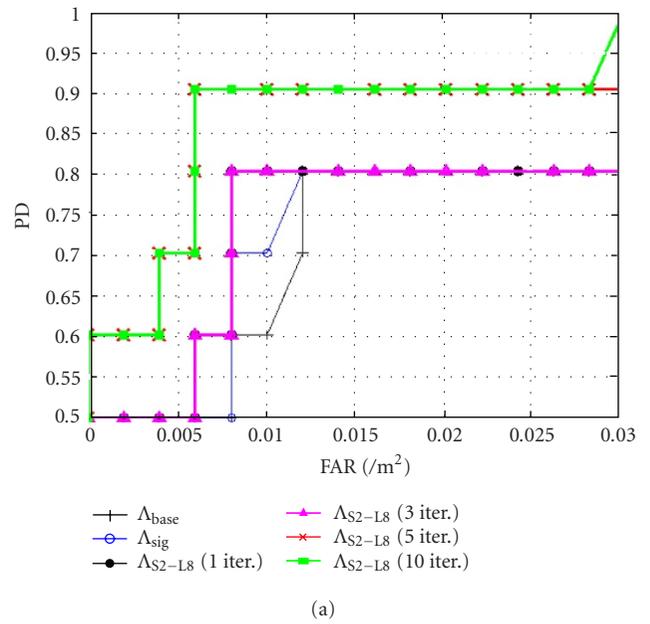
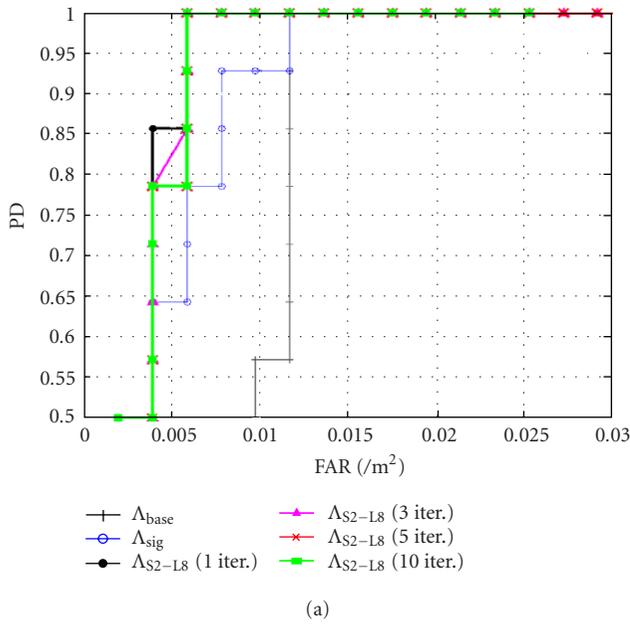


FIGURE 10: Performance of the HMM detector on one pass from lane 4. (a) ROCs with and without corrective training. (b) Improvement in the performance (area under ROC in the $[0, 0.02]$ FAR range) versus number of corrective training iterations.

FIGURE 11: Performance of the HMM detector on one pass from lane 6. (a) ROCs with and without corrective training. (b) Improvement in the performance (area under ROC in the $[0, 0.02]$ FAR range) versus number of corrective training iterations.

We have also grouped the passes by lanes, and generated one ROC for each lane. The performance of the different lanes is shown in Figure 12(a). As can be seen, the performance varies significantly. Lane 4 has an almost perfect performance, while lane 9 has the worst performance. This type of performance is expected since lane 4 contains mainly metal mines, while lane 9 contains mainly small plastic-cased mines. Lanes 6 and 8 contain a mixture of both types of mines. For all the lanes, the improvement due to the corrective training is not obvious in Figure 12(a). This is due to the relatively large variation in the vertical axis. To outline the effect of the corrective training, in Figure 12(b), we display the areas obtained with the corrective training mod-

els normalized with respect to the area of the corresponding baseline system. As can be seen, the improvement varies from 4% to 27%. The best improvement is obtained for lane 9, which was the most difficult lane. This illustrates the ability of the proposed corrective training to adapt the models' parameters and improve the detection of difficult mines.

Data collection 2 has a total of 83 passes that were collected over three different dates. The first 20 passes were collected on April 8 and 9, the next 27 passes were collected on June 11 and 12, and the last 36 passes were collected on June 27 and 28. On each date, data was collected from all 4 lanes. We have generated one ROC for each date using the different

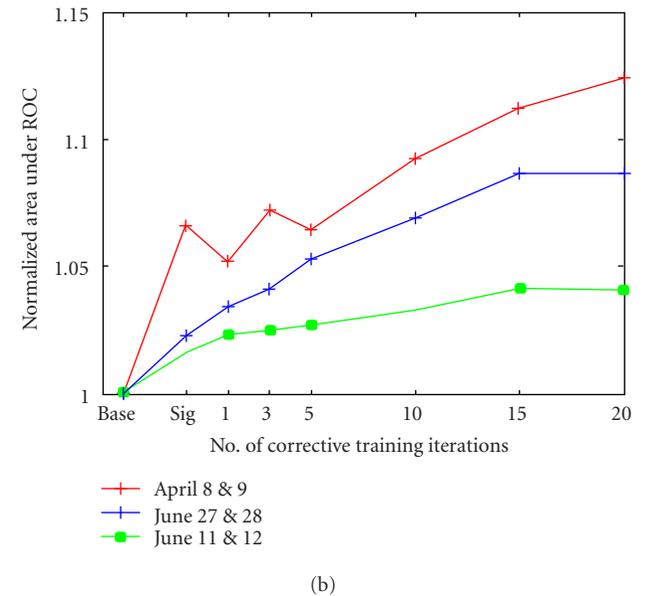
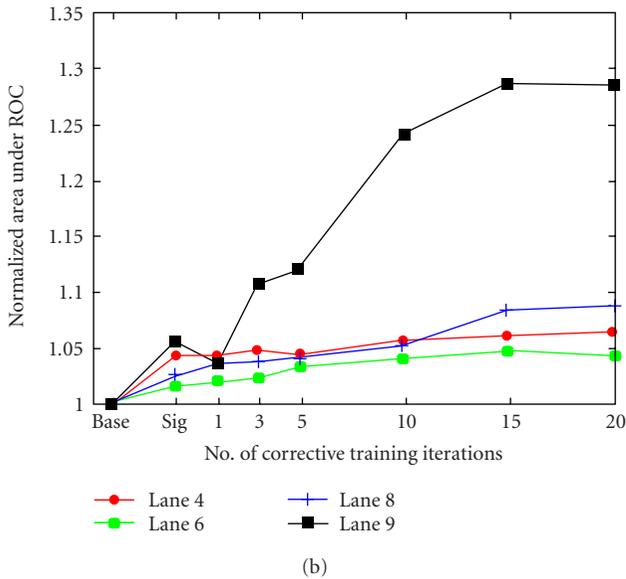
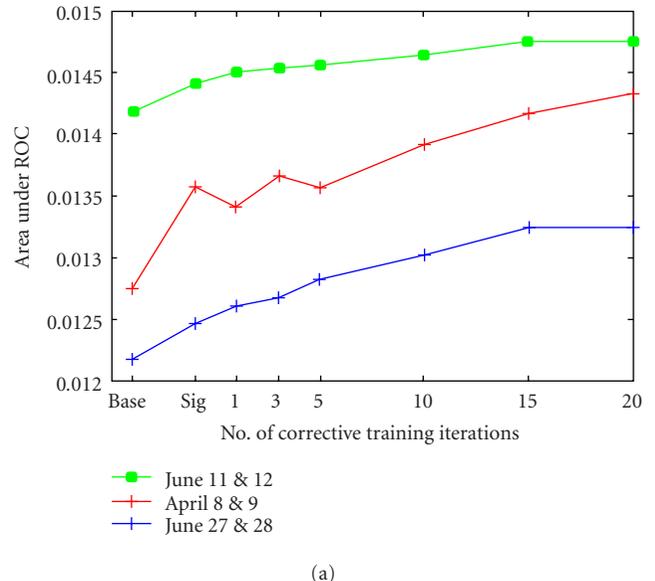
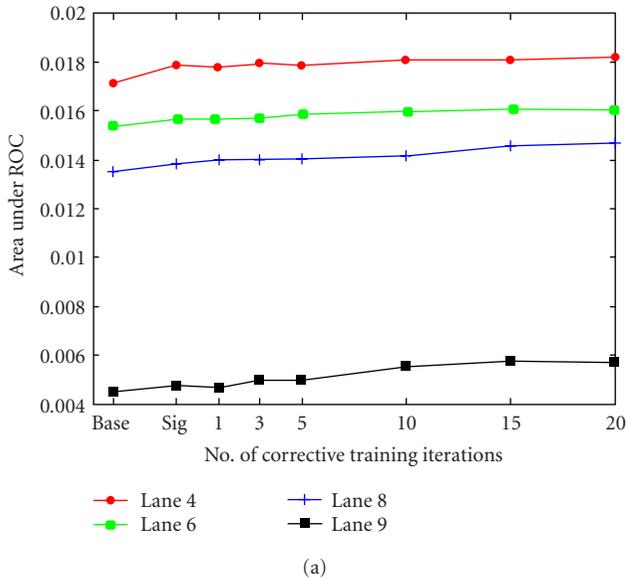


FIGURE 12: Average performance of the HMM detector on collection 2 grouped by lanes. (a) Area under ROCs in the $[0, 0.02]$ FAR range versus number of corrective training iterations. (b) Normalized areas under ROC in the $[0, 0.02]$ FAR range with respect to the area of the baseline system.

FIGURE 13: Average performance of the HMM detector on collection 2 grouped by dates. (a) Area under ROCs in the $[0, 0.02]$ FAR range versus number of corrective training iterations. (b) Normalized areas under ROC in the $[0, 0.02]$ FAR range with respect to the area of the baseline system.

HMMs. Figure 13(a) displays the performance on each date. As can be seen, even though the same lanes were used, the performance on the different dates varies. This is due mainly to the different weather and soil conditions, and possibly different radar settings, on the different dates. Figure 13(b) displays the normalized areas to emphasize the improvements due to the corrective training. The improvement varies from 4% to 12.5%. It is interesting to notice that, in this case, the best improvement (April data) does not correspond to the date that had the worst performance (June 27 and 28 data). This can be explained by the fact that the pass used for the corrective training was collected in April also. Thus, these re-

sults show that the corrective training could be used to adapt the model parameters to the different soil and environment conditions.

The above experiments were repeated using collection 3 and the HMMs obtained by performing corrective training using one pass from lane 51 of this collection ($\Lambda_{S3-L51}(k)$, $k = 1, \dots, 20$). The results and conclusions were comparable to those obtained for collection 2. Figure 14(a) shows the overall ROCs for all of the 18 passes of collection 2. The ROCs are displayed for the confidence values generated using the baseline model, Λ_{base} , and $\Lambda_{S3-L51}(10)$ models. No significant improvement can be noticed using models learned with

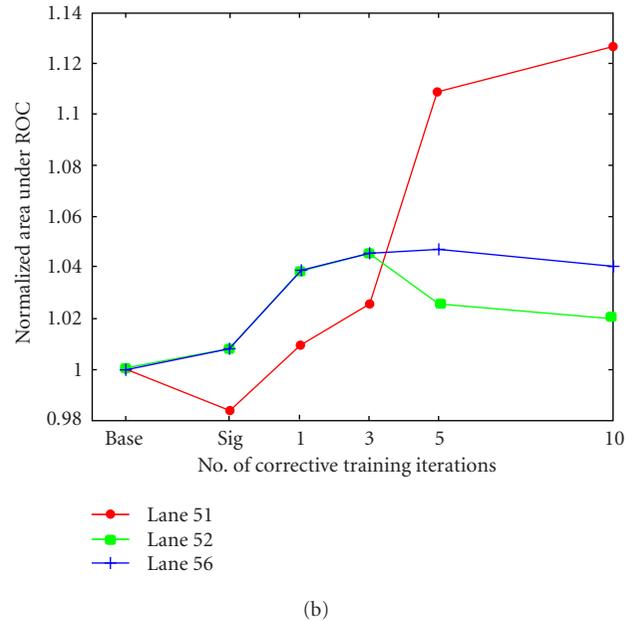
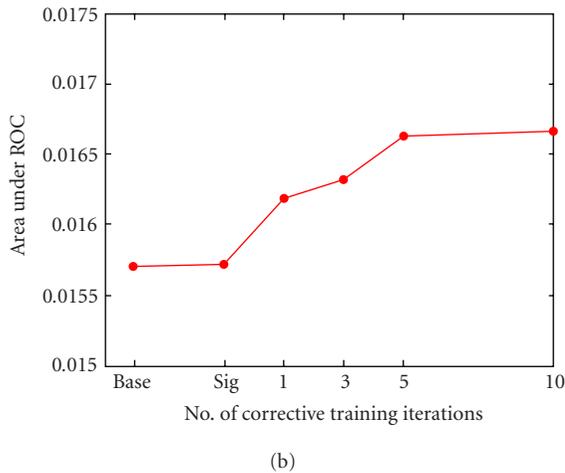
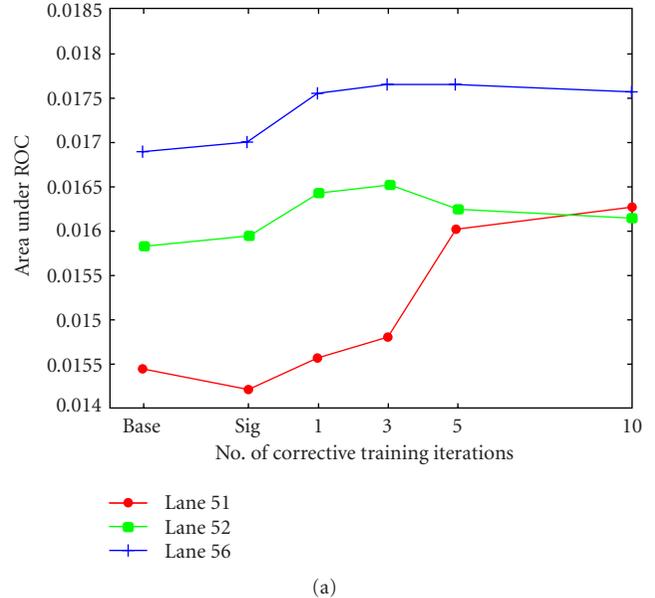
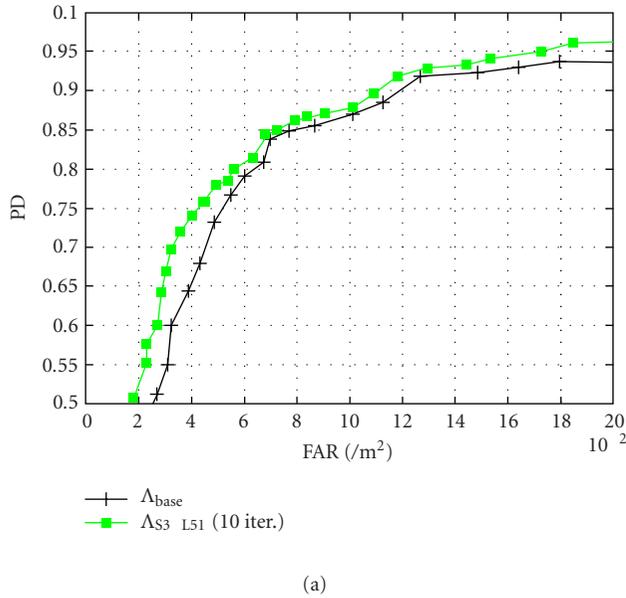


FIGURE 14: Average performance of the HMM detector with different model parameters on the 18 passes of collection 3. (a) ROCs with and without corrective training. (b) Improvement in the performance (area under ROC in the [0, 0.02] FAR range) versus number of corrective training iterations.

more than 10 iterations of corrective training. Figure 14(b) shows the area under the ROC in the [0, 0.02] FAR range. As can be seen, after 5 iterations, the overall performance improves by about 8%.

Figure 15 displays the performance grouped by lanes. As in collection 2, since the lanes have different mixtures of mine types, their performance varies. Lane 51 benefited most from the corrective training. This is due to the fact that this lane had the worst initial performance, so there is more room for improvement. Also, the pass that was used for corrective training was extracted from this lane. Thus, the trained models were adapted to this lane. It is interesting to notice that the performance on lane 51 using Λ_{sig} is slightly worse than the performance using Λ_{base} . This may be due to the fact that the data from this lane has different characteristics than the

FIGURE 15: Average performance of the HMM detector on collection 3 grouped by lanes. (a) Area under ROCs in the [0, 0.02] FAR range versus number of corrective training iterations. (b) Normalized areas under ROC in the [0, 0.02] FAR range with respect to the area of the baseline system.

signature library. Thus, a model that minimizes the misclassification of the signatures may not be suited for a site with different characteristics.

6. CONCLUSIONS

A real-time software system for landmine detection using ground-penetrating radar is proposed and evaluated.

The system includes three main components: preprocessing; HMM-based detector; and corrective training. The preprocessing component, which is needed to enhance the mine signatures, requires the estimation of subpixel shift and scale from one vector sample to the next. To reduce the computational requirements and facilitate real-time implementation, we proposed an efficient frequency-domain-based processing. The HMM detector is an improvement of a previously proposed system (baseline). It includes additional pre- and postprocessing steps of the observation sequences to improve the time efficiency and enable real-time application. It also treats the sensors' channels independently, and uses one background model per channel. This allows parallel processing of the different channels and better adaptation. The corrective training component is used to adjust the initial model parameters to minimize the number of misclassification sequences. Two corrective training scenarios were used. The first one is offline, and is applied to a signature library to adjust the parameters of a generic model. The second scenario mimics a real-world operational mode where, feedback on which measurements are mines and false alarms once they are dug, would be used to adapt the initial models to specific sites and environments.

The proposed system was tested in real-time settings. Extensive experiments indicate that the corrective training can improve the overall performance by about 10%, and that for some individual lanes, the performance gain can reach up to 50%. Moreover, the best gain in performance is usually obtained for the difficult lanes that had low mine detection and high false-alarm rates using the baseline parameters.

In the current system, one pass (from one lane) was used to adapt the model parameters to all the data collected from one site. In the future work, we will experiment with adapting the parameters to individual lanes using a small calibration segment from each lane. Another direction for future research involves building and learning different models for the different mine types (e.g., plastic versus metal). This may require extracting multiple feature sets, and using different subsets to characterize the different models.

Finally, we should note that our adaptive system can learn to discriminate between mines and clutter objects as long as they have different signatures. However, some clutter objects may have GPR signatures very similar to mine signatures. For instance, a 5 cm thick metal disk with a 30 cm diameter, or a metal can may be incorrectly classified as mines. In this case, different features and classifiers, or even different sensors, may be needed.

ACKNOWLEDGMENTS

The work reported on here was supported in part by the US-Army-Research-Office-funded Multidisciplinary Research on Mine Detection and Neutralization Systems, Contract no. DAAG55-97-1-0014, and by the US Army Grant nos. DAAB15-01-D-0004 and DAAB15-02-D-0003. We thank Dr. James Harvey and Mr. Pete Howard for their support of this research. We also thank the reviewers for their constructive comments.

REFERENCES

- [1] Landmines, Mine Action News from the United Nations, vol. 3.2, 4th Quarter, 1998.
- [2] "Hidden Killers: The Global Landmine Crisis," United States Department of State Report, Publication no. 10575, September 1998.
- [3] T. R. Witten, "Present state of the art in ground-penetrating radars for mine detection," in *Detection and Remediation Technologies for Mines and Minelike Targets III*, vol. 3392 of *Proceedings of SPIE*, pp. 576–585, Orlando, Fla, USA, April 1998.
- [4] P. D. Gader, H. Frigui, B. N. Nelson, G. Vaillette, and J. M. Keller, "New results in fuzzy-set-based detection of land mines with GPR," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 1075–1084, Orlando, Fla, USA, April 1999.
- [5] H. T. Haskett and J. T. Broach, "Automatic mine detection algorithm using ground penetration radar signatures," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 942–952, Orlando, Fla, USA, April 1999.
- [6] D. Carevic, "Clutter reduction and target detection in ground-penetrating radar data using wavelets," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 973–978, Orlando, Fla, USA, April 1999.
- [7] D. Carevic, "Kalman filter-based approach to target detection and target-background separation in ground-penetrating radar data," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 1284–1288, Orlando, Fla, USA, April 1999.
- [8] A. H. Gunatilaka and A. B. Baertlein, "Subspace decomposition technique to improve GPR imaging of antipersonnel mines," in *Detection and Remediation Technologies for Mines and Minelike Targets V*, vol. 4038 of *Proceedings of SPIE*, pp. 1008–1018, Orlando, Fla, USA, April 2000.
- [9] H. Brunzell, "Detection of shallowly buried objects using impulse radar," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, no. 2, pp. 875–886, 1999.
- [10] S.-H. Yu, R. K. Mehra, and T. R. Witten, "Automatic mine detection based on ground penetrating radar," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 961–972, Orlando, Fla, USA, April 1999.
- [11] H. Frigui, P. D. Gader, and K. Satyanarayana, "Landmine detection with ground penetrating radar using fuzzy k-nearest neighbors," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ '04)*, vol. 3, pp. 1745–1749, Budapest, Hungary, July 2004.
- [12] P. D. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, no. 6, pp. 1231–1244, 2001.
- [13] S. L. Tantum, Y. Wei, V. S. Munshi, and L. M. Collins, "Comparison of algorithms for land mine detection and discrimination using ground penetrating radar," in *Detection and Remediation Technologies for Mines and Minelike Targets VII*, vol. 4742 of *Proceedings of SPIE*, pp. 728–735, Orlando, Fla, USA, April 2002.
- [14] J. M. Stiles, P. Parra-Bocaranda, and A. Apte, "Detection of object symmetry using bistatic and polarimetric GPR observations," in *Detection and Remediation Technologies for Mines and Minelike Targets IV*, vol. 3710 of *Proceedings of SPIE*, pp. 992–1002, Orlando, Fla, USA, April 1999.

- [15] H. Frigui, K. Satyanarayana, and P. D. Gader, "Detection of land mines using fuzzy and possibilistic membership functions," in *Proc. 12th IEEE International Conference on Fuzzy Systems (FUZZ '03)*, vol. 2, pp. 834–939, Saint Louis, Mo, USA, May 2003.
- [16] P. D. Gader, B. N. Nelson, H. Frigui, G. Vaillette, and J. M. Keller, "Fuzzy logic detection of landmines with ground penetrating radar," *Signal Processing*, vol. 80, no. 6, pp. 1069–1084, 2000.
- [17] P. D. Gader, R. Grandhi, W.-H. Lee, J. N. Wislon, and D. K. C. Ho, "Feature analysis for the NIITEK ground-penetrating radar using order-weighted averaging operators for landmine detection," in *Detection and Remediation Technologies for Mines and Minelike Targets IX*, vol. 5415 of *Proceedings of SPIE*, pp. 953–962, Orlando, Fla, USA, April 2004.
- [18] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1996.
- [19] Y. Zhao, P. D. Gader, P. Chen, and Y. Zhang, "Training DHMMs of mine and clutter to minimize landmine detection errors," *IEEE Trans. Geosci. Remote Sensing*, vol. 41, no. 5, pp. 1016–1024, 2003.
- [20] S. Mizuta and K. Nakajima, "An optimal discriminative training method for continuous mixture density HMMs," in *Proc. 1st International Conference on Spoken Language Processing (ICSLP '90)*, pp. 245–248, Kobe, Japan, November 1990.
- [21] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [22] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [23] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.
- [24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [25] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal (ICASSP '86)*, pp. 49–52, Tokyo, Japan, April 1986.
- [26] B. Meriardo, "Phonetic recognition using hidden Markov models and maximum mutual information training," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal (ICASSP '88)*, vol. 1, pp. 111–114, New York, NY, USA, April 1988.
- [27] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.
- [28] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, 1998.
- [29] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "A new algorithm for the estimation of hidden Markov model parameters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal (ICASSP '88)*, vol. 1, pp. 493–496, New York, NY, USA, April 1988.
- [30] J. R. R. Pressley, D. Pabst, G. D. Sower, L. Nee, B. Green, and P. Howard, "Ground standoff mine detection system (GSTAMIDS) engineering, manufacturing, and development (EMD) Block 0," in *Detection and Remediation Technologies for Mines and Minelike Targets VI*, vol. 4394 of *Proceedings of SPIE*, pp. 1190–1200, Orlando, Fla, USA, April 2001.
- [31] G. D. Sower, J. Eberly, and E. Christy, "GSTAMIDS ground-penetrating radar: hardware description," in *Detection and Remediation Technologies for Mines and Minelike Targets VI*, vol. 4394 of *Proceedings of SPIE*, pp. 651–661, Orlando, Fla, USA, April 2001.
- [32] E. G. Farr and C. E. Baum, "Prepulse associated with the TEM feed of an impulse radiating antenna," *Sensor and Simulation Note 37*, Air Force Research Laboratory, March 1992.
- [33] E. G. Farr, "Optimizing the feed impedance of impulse radiating antennas—Part I: Reflector IRAs," *Sensor and Simulation Note 354*, Air Force Research Laboratory, January 1993.
- [34] E. G. Farr and C. J. Buchenauer, "Experimental validation of IRA models," *Sensor and Simulation Note 364*, Air Force Research Laboratory, January 1994.
- [35] T. Kohonen, "Improved versions of learning vector quantization," in *Proc. International Joint Conference on Neural Networks (IJCNN '90)*, vol. 1, pp. 545–550, San Diego, Calif, USA, June 1990.
- [36] J. R. R. Pressley, D. Pabst, G. D. Sower, L. Nee, B. Green, and P. Howard, "Ground standoff mine detection system (GSTAMIDS) engineering, manufacturing and development (EMD) block 0," in *Detection and Remediation Technologies for Mines and Minelike Targets VI*, vol. 4394 of *Proceedings of SPIE*, pp. 1190–1200, Orlando, Fla, USA, April 2001.

Hichem Frigui is an Assistant Professor and the Director of the Multimedia Research Lab, the University of Louisville, Kentucky. He received his Ph.D. degree in computer engineering and computer science from the University of Missouri, Columbia, in 1997. From 1998 to 2004, he was an Assistant Professor at the University of Memphis. He has been active in the research fields of pattern recognition, data mining, image processing, and content-based image retrieval. He has published over 60 journal and refereed conference articles, and has received the National Science Foundation Career Award for outstanding young scientists. Dr. Frigui has been active in landmine detection algorithm research since 1998. He was a member of few teams that developed, implemented, and field-tested several real-time algorithms for mine detection using ground penetrating radar. Dr. Frigui is a Member of the IEEE, IEEE Computer Society, and ACM. He is currently serving as an Associate Editor of the *IEEE Transactions on Fuzzy Systems*, and the *International Journal of Fuzzy Systems*.



K. C. Ho was born in Hong Kong. He received the B.S. degree with first-class honours in electronics and the Ph.D. degree in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 1988 and 1991, respectively. He was a Research Associate in the Royal Military College of Canada from 1991 to 1994. He joined the Bell-Northern Research, Montreal, Canada, in 1995, as a member of the scientific staff. He was a faculty member at the University of Saskatchewan, Saskatoon, Canada, from September 1996 to August 1997. Since September 1997, he has been with the University of Missouri, Columbia, where he is currently an Associate Professor in the Electrical and Computer Engineering Department. His research interests are in statistical and adaptive signal processing, sub-surface object detection, source localizations, wavelet transform, and wireless communications. Dr. Ho is currently serving as an



Associate Editor of the IEEE Transactions on Signal Processing, and the IEEE Signal Processing Letters. He is also the Editor of the *ITU Standard Recommendation G.168: Digital Network Echo Cancellers*. He has three US patents, three Canadian patents, and three European patents in the area of mobile communications.

Paul Gader received his Ph.D. degree in mathematics in 1986 from the University of Florida. He has worked as a Senior Research Scientist at Honeywell's Systems and Research Center, as a Research Engineer and Manager at the Environmental Research Institute of Michigan, and as a faculty member at the University of Wisconsin, the University of Missouri, and the University of Florida, where he is currently a Professor of



computer and information science and engineering. He led teams involved in real-time, handwritten address recognition systems for the US postal service developing algorithms for handwritten digit recognition and segmentation, numeric field recognition, word recognition, and line segmentation. He has led teams that devised and tested several real-time algorithms in the field for mine detection. He served as a Technical Director of the University of Missouri MURI on Humanitarian Demining for two years. He is currently involved in landmine detection projects investigating handheld and ground-based detection systems, acoustic detection, EO/IR detection of mines and trip-wires, hyperspectral detection, and multi-sensor fusion. Dr. Gader is a Senior Member of the IEEE and has over 165 technical publications in the areas of image and signal processing, applied mathematics, and pattern recognition, including over 50 refereed journal articles.