Training DHMMs of Mine and Clutter to Minimize Landmine Detection Errors

Yunxin Zhao, *Senior Member, IEEE*, Paul Gader, *Senior Member, IEEE*, Ping Chen, and Yue Zhang

***Abstract*—Minimum classification error (MCE) training is proposed to improve performance of a discrete hidden Markov model (DHMM)-based landmine detection system. The system (baseline) was proposed previously for detection of both metal and nonmetal mines from ground-penetrating radar signatures collected by moving vehicles. An initial DHMM model is trained by conventional methods of vector quantization and the Baum–Welch algorithm. A sequential generalized probabilistic descent (GPD) algorithm that minimizes an empirical loss function is then used to estimate the landmine/background DHMM parameters, and an evolutionary algorithm (EA) based on fitness score of classification accuracy is used to generate and select codebooks. The landmine data of one geographical site was used for model training, and those of two different sites were used for evaluation of system performance. Three scenarios were studied: 1) apply MCE/GPD alone to DHMM estimation, 2) apply EA alone to codebook generation, and 3) first apply EA to codebook generation and then apply MCE/GPD to DHMM estimation. Overall, the combined EA and MCE/GPD training led to the best performance. At the same level of detection rate as the baseline DHMM system, the false-alarm rate was reduced by a factor of two, indicating significant performance improvement.**

***Index Terms*—Discrete hidden Markov model (DHMM), evolutionary algorithm (EA), generalized probabilistic descent (GPD), ground-penetrating radar (GPR) signature, hidden Markov model (HMM), landmine detection, minimum classification error (MCE) training.**

I. Introduction

LANDMINES that are buried in war zones around the world continue to pose threat to the safety of civilians and soldiers [1]. Detection and removal of landmines is therefore a significant problem and has attracted active research efforts 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. Ground-penetrating radar (GPR) is known to provide informative responses from both metal and nonmetal mines, and it has attracted significant efforts on GPR data-based landmine detection [2]–[12]. However, GPR signals vary greatly with the sizes and types of mines, soil, and weather conditions, etc. As the result, a high rate of mine detection is often accompanied by a high rate of false alarm. 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 characteristic (ROC) curve that jointly specifies rate of mine detection and level of false alarm.

In GPR-based landmine detection, various competing approaches have been proposed for feature analysis and classifier design. Feature analysis approaches include principle component analysis [5], wavelets [6], image processing methods of derivative feature extraction [10], and curve analysis using Hough and Radon transforms [12], as well as model-based methods [11]. Classifier design approaches include Bayesian [12], hidden Markov models [10], fuzzy logic [9], etc. In [10], discrete hidden Markov modeling (DHMM or HMM) was proposed for detection of landmines of both metal and nonmetal types using images collected by a moving-vehicle-mounted GPR system. The HMM-based landmine detection system consists of preprocessing GPR images to remove stationary artifacts, extracting features to represent the parabolic shapes of landmine GPR signatures, training codebook and HMMs of mine and background, and computing confidence scores of mine and background by the respective HMMs. Experimental evaluation results support the notion that HMM is feasible and effective for landmine detection. In comparison with methods of energy detector and fuzzy gradient-based automatic target recognizer, the HMM technique performed significantly better than the former and achieved comparable performance to the latter [10].

In this paper, an improvement to the HMM system of [10] is proposed in the aspect of model training. In [10], the training of mine and background models was based on the conventional concept of distribution fitting: the codebook was generated by nearest neighbor clustering-based vector quantization (VQ) that minimizes the distortion between observation data and codebook [13], and the HMM parameters were estimated by the Baum–Welch (BW) algorithm, which maximizes the likelihood of observation data [14]. However, neither VQ nor BW has direct impacts on classification errors. As shown in the field of speech recognition [15], [16], by changing the model training criterion from distribution fitting to direct minimization of classification errors, accuracy of statistical classifiers can be significantly improved. This paper extends this concept to the training of mine and background models. Specifically, the minimum classification error (MCE) and generalized probabilistic descent (GPD) training is proposed for estimation of DHMM parameters, and an evolutionary algorithm (EA) [17] is proposed for codebook generation, where both MCE/GPD and EA adjust model parameters to minimize classification errors on training data.

Manuscript received August 9, 2002; revised February 14, 2003. This work was supported by the OSD-funded Multi-Disciplinary Research on Mine Detection and Neutralization Systems under Contract DAAG559710014.

Y. Zhao, P. Chen, and Y. Zhang are with the Department of Computer Engineering and Computer Science, University of Missouri, Columbia, MO 65211 USA (e-mail: zhaoy@missouri.edu).

P. Gader is with the Deptartment of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA.

Digital Object Identifier 10.1109/TGRS.2003.811761

The MCE/GPD training technique has been extensively studied in the area of speech recognition. The key of the MCE method is the formulation of an error function that incorporates recognition operation and performance of an arbitrary classifier in a functional form, called the loss function, which can be directly evaluated and optimized [15], [16]. The loss function is commonly taken as a sigmoid function of a misclassification measure that is derived from the decision rule of the classifier. MCE training attempts to estimate model parameters that minimize the expected or empirical loss function over a set of training data by probabilistic descent. One important consequence of MCE training is that the classifier's performance becomes less affected by the often-present mismatch of assumed and true distributions of data, since the objective of training is no longer focused on fitting distributions to training data. In fact, MCE/GPD is capable of minimizing classification errors on training data for arbitrary choice of distributions or discriminant functions. In speech recognition, MCE/GPD training has been mainly applied to continuous density HMMs, and it was shown effective in recognition of highly confusable speech sounds and in utterance verification [18].

EAs are a class of stochastic search algorithms for global optimization problems. The EAs are population based, and in a population, there are communications and information exchange among individuals based on the outcomes of selection, competition and recombination. Depending on different search operators and selection mechanisms, the EAs can be categorized into three major branches: genetic algorithm (GA), evolutionary programming (EP) and evolution strategies (ESes) [19]. In terms of search operators, GA normally uses crossover and mutation, EP uses only mutation, and ES uses recombination and mutation. In terms of selection mechanisms, GA and EP are probabilistic while ES is deterministic. In addition, GA encodes individuals by binary strings, while EP and ES use real-valued vectors directly as individuals. In [20], the concept of MCE was applied to discriminative codebook training in a DHMM-based system for recognition of highly confusable words. Due to the complicated optimization surface involved in VQ, the GPD algorithm was difficult to apply and an ES algorithm was employed instead. In the ES-based codebook training, the DHMM parameters were held fixed, and successive modifications and selections of codewords were performed to minimize classification errors. A "fast EP" was proposed in [21], where Cauchy instead of Gaussian mutation was used as the primary search operator to enable longer jumps when search points were far away from the global optimal point.

The current work proposes an extension of MCE training to DHMMs on both codebook and DHMM parameters, and investigates its application to landmine detection. The EP method of [21] was adapted to codebook training to take advantage of its convergence property. The landmine and background models trained by conventional methods were used as the initial models to start the discriminative training. The same GPR data as used in [10] were used for training and evaluation of the proposed system, and the results were compared with those in [10].

This paper is organized in four sections. In Section II, the training algorithms of MCE/GPD and EP for landmine and background models are described. In Section III, the experimental data, procedure, and results are discussed and summarized. Conclusions are provided in Section IV.

II. Model Training for Classification Error Minimization

Discrete hidden Markov modeling of mine and background consists of a codebook V and two sets of HMM parameters $\lambda^{(l)}$, with $l = 1$ for mine and $l = 2$ for background. The codebook consists of M codewords $V = \{v_1, v_2, \ldots, v_M\}$, and the codewords are representative patterns of mine or background observation feature vectors. Each HMM has N states, with $\lambda^{(l)}$ consisting of state-transition probabilities $a_{ij}^{(l)}$, state-initial probabilities $\pi_i^{(l)}$, and state-conditional symbol emission probabilities $b_{jk}^{(l)}$, where $i, j = 1, 2, \ldots, N$, $k = 1, 2, \ldots, M$. Let $\Lambda = \{\lambda^{(1)}, \lambda^{(2)}\}$. Given a certain amount of mine and background training data, the task of model training is to obtain the codebook V and the HMM parameters Λ according to a certain optimization criterion.

A. MCE/GPD Training of DHMM Parameters

Denote an observation sequence of mine or background by $O = (o_1, o_2, \ldots, o_T)$, with o_t a feature vector obtained at a location t. The quantization operation on an observation vector is defined as $Q_V(o_t) = \arg\min_{1\le k\le M} d(o_t, v_k)$, where $d()$ is a distance measure. In this paper, $d()$ is taken as the Euclidean distance. Given HMM parameters $\lambda^{(l)}$, the joint probability of observation O and its underlying state sequence q is

$$P\left(O, q | V, \lambda^{(l)}\right) = \pi_{q_o}^{(l)} \prod_{t=1}^{T} a_{q_{t-1}q_t}^{(l)} b_{q_t Q_V(o_t)}^{(l)}.$$

The discriminant function for classifying O to one of the two classes $\{C_1, C_2\} = \{\text{mine}, \text{background}\}$ is taken as the log of the joint probability of the observation and its most likely underlying state sequence, i.e.,

$$g_l(O; V, \Lambda) = \max_q \log P\left(O, q | V, \lambda^{(l)}\right)$$

where the max operation can be efficiently computed by a Viterbi algorithm [14]. The classifier's decision rule $\mathcal{C}$ is

$$\mathcal{C}(O) = C_l \text{ if } g_l(O; V, \Lambda) = \max_{l'} g_{l'}(O; V, \Lambda).$$

If $O \in C_l$ but $g_l(O; V, \Lambda) < g_{l'}(O; V, \Lambda)$, $l \neq l'$, then a misclassification occurs. Accordingly, a misclassification measure is defined for class l as

$$d_l(O; V, \Lambda) = -g_l(O; V, \Lambda) + g_{l'}(O; V, \Lambda), \qquad l \neq l'$$

where a positive d_l value indicates misclassification, and a negative d_l value indicates correct classification. The misclassification measure is further embedded into a 0-1 sigmoid function, referred to as loss function

$$\mathcal{L}_l(O; \Lambda) = \frac{1}{1 + \exp\left(-\gamma d_l(O; V, \Lambda)\right)}.$$

Correct classification corresponds to loss values in [0, 1/2), and misclassification corresponds to loss values in (1/2, 1]. An equivocal case occurs when $d_l = 0$ or $\mathcal{L}_l = 1/2$. The shape of the sigmoid loss function varies with the parameter $\gamma > 0$: the larger the γ, the narrower the transition region.

Given a set of training observation sequences O_r, $r = 1, 2, \ldots, R$, an empirical loss function on the training dataset is defined as

$$\mathbf{L}(\Lambda) = \sum_{r=1}^{R} \sum_{l=1}^{2} \mathcal{L}_l(O_r; \Lambda) 1(O_r \in C_l)$$

where $1(O_r \in C_l)$ is an indicator function that takes the value of 1 if $O_r \in C_l$ and 0 otherwise. Minimizing the empirical loss is equivalent to minimizing total misclassification errors. The HMM parameters are therefore estimated by carrying out a gradient descent on $\mathbf{L}(\Lambda)$. In order to ensure that the estimated HMM parameters satisfy the stochastic constraints of $a_{ij} \geq 0$, $\sum_{j=1}^{N} a_{ij} = 1$ and $b_{jk} \geq 0$, $\sum_{j=1}^{M} b_{jk} = 1$, parameter mappings of $a_{ij} \rightarrow \tilde{a}_{ij}$ and $b_{jk} \rightarrow \tilde{b}_{jk}$ are performed by $a_{ij} = (\exp(\tilde{a}_{ij}))/(\sum_{j'=1}^{N} \exp(\tilde{a}_{ij'}))$ and $b_{jk} = (\exp(\tilde{b}_{jk}))/(\sum_{k'=1}^{M} \exp(\tilde{b}_{jk'}))$, and the transformed HMM parameters are denoted by $\tilde{\Lambda}$. Parameter updates are performed with respect to $\tilde{\Lambda}$. Adopting a sequential estimation mode, the HMM parameters are updated for successive observations O_r as

$$\tilde{\Lambda}(r+1) = \tilde{\Lambda}(r) - \frac{\epsilon}{r} \nabla_{\tilde{\Lambda}} \sum_{l=1}^{2} \mathcal{L}_l(O_r; \tilde{\Lambda})\, 1(O_r \in C_l)|_{\tilde{\Lambda} = \tilde{\Lambda}(r)}, \quad r = 1, 2, \ldots, R \tag{1}$$

where ϵ controls the estimation step size. Details of the gradient equations are provided in the Appendix. The initial HMM parameters $\Lambda(0)$ or $\tilde{\Lambda}(0)$ are taken as those trained by the BW algorithm. Define one training iteration as applying (1) on the full set training sequences O_r, $r = 1, 2, \ldots, R$. Multiple training iterations are performed on the training dataset, where the HMM parameters obtained at the end of the current iteration are used as the initial HMM parameters in the next iteration. The training process terminates if the empirical loss converges or the number of iterations reaches a preset limit. The parameters γ and ϵ are empirically chosen in the experiments.

B. EP Training of Codebooks

An initial population of parent codebooks are generated from a codebook that is trained by a conventional VQ algorithm. The codebook training then goes through iterations of producing offsprings and selecting next-generation parents. The offspring production is based on mutation, and parent selection is based on fitness score of correct classifications on a training dataset.

Initialization of Parent Codebooks: Given a codebook V, G parent codebooks $V_1^{(1)}, V_2^{(1)}, \ldots, V_G^{(1)}$ are first generated by perturbations on V. Let the codewords of a codebook be stacked into a long vector. The codebook $V_i^{(1)}$ and an accompanied random vector $\eta_i^{(1)}$, together referred to as the ith individual of the first iteration, are generated in each dimension j as

$$\begin{aligned} V_i^{(1)}(j) &= V(j) + \eta^{(0)} \delta_{ij} \\ \eta_i^{(1)}(j) &= \eta^{(0)} \exp(\tau' \nu_i + \tau \nu_{ij}) \end{aligned}$$

where $\eta^{(0)}$ is initialized as a constant; δ_{ij} is a zero-mean unit-scale Cauchy random variable; ν_i and ν_{ij} are both $\mathcal{N}(0, 1)$; and the constants τ and τ' are set as $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$ as in [17] and [22], with n the dimension of the stacked codebook vector.

Generation of Offspring Codebooks: In the kth iteration, $k = 1, 2, \ldots$, offspring individuals are generated from parent individuals, with one parent producing one offspring. From the parent individuals $(V_i^{(k)}, \eta_i^{(k)})$, $i = 1, 2, \ldots, G$, the offspring individuals $(V_i^{(k)}, \eta_i^{(k)})$, $i = G+1, \ldots, 2G$, are generated by

$$\begin{aligned} V_i^{(k)}(j) &= V_{i-G}^{(k)}(j) + \eta_{i-G}^{(k)}(j) \delta_{ij} \\ \eta_i^{(k)}(j) &= \eta_{i-G}^{(k)}(j) \exp(\tau' \nu_i + \tau \nu_{ij}) \end{aligned}$$

where the random variables δ_{ij} and ν_{ij} are generated anew for each i and j and for each iteration k, and ν_i is generated anew for each i and each k.

Selection of Next-Generation Codebooks: The fitness score of the individual $(V_i^{(k)}, \eta_i^{(k)})$ is denoted by F_i, $i = 1, 2, \ldots, 2G$. In computing F_i, the codebook $V_i^{(k)}$ is paired with the HMM parameters $\mathbf{\Lambda}$ to classify the training GPR signatures of mine and background. For each individual i, G' opponents are chosen randomly from the $2G$ individuals with equal probability, and the opponent set is denoted by $\Omega_{\bar{i}}$. The fitness score F_i is compared with the fitness scores of the opponents F_j, $j \in \Omega_{\bar{i}}$, and the number of wins by F_i, denoted by w_i, is counted as

$$w_i = \#(F_i > F_j), \qquad j \in \Omega_{\bar{i}}.$$

The counts w_i, $i = 1, 2, \ldots, 2G$, are sorted in an descending order and the top G individuals are selected as next generation codebooks, denoted by $(V_i^{(k+1)}, \eta_i^{(k+1)})$, $i = 1, 2, \ldots, G$. These codebooks become parents for offspring production in the $(k+1)$th iteration.

Termination: In each iteration, the average fitness score of the selected top G individuals is computed. If in two consecutive iterations the relative change of the average fitness scores drops below a predefined threshold, convergence is concluded and the training is terminated. The number of training iterations can also be limited by an empirical number. The codebooks of the selected top G individuals in the last iteration are taken as the EP-optimized codebooks.

III. Experiments

A. Experimental Data

The GPR data collected from three sites of the United States were used in the experiments: temperate site 1, temperate site 2, and arid site 1. The data of temperate site 1 were organized into a signature library, consisting of 2945 landmine and 5737 background observation sequences that corresponded to 636 landmine and 463 background signatures. The library was used to train the landmine and background HMM models. The GPR lane data of arid site 1 and temperate site 2 were each divided into two sets: calibration set and blind test set. Previously, the locations of the mines on the blind test lanes were not known. These lanes were used for field testing several years ago. The locations have since been released and the data acquired from

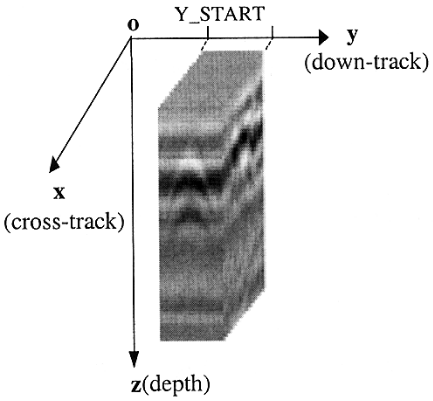


Fig. 1. Illustration of GPR landmine signature.

both sets of lanes can now be used for algorithm development and evaluation. On the arid site 1, there were 12 calibration lanes and 16 blind test lanes. On the temperate site 2, there were 16 calibration lanes and six blind test lanes. Each lane was 3 m wide and the length of lanes varies from 50–200 m. Multiple data files were collected for each lane, with each file corresponding to running a vehicle-mounted GPR system over the lane once, referred to as one pass. It is worth noting that arid site 1 and temperate site 2 both have more than 50% plastic-cased mines. A more detailed description on data of the three sites can be found in [10].

B. Feature Extraction

A typical landmine signature is shown in Fig. 1, where x, y, and z axes represent the directions of down-track, cross-track, and depth, respectively. The unit of y is in scans, where a scan is generated for every 5 cm distance. The unit of x is in pixels, one pixel corresponding to 4.29 cm. The unit of z is also in pixels, the total number of pixels is 30 but the penetration depth of GPR varies. In the y–z plane, a landmine is seen to have a parabolic shape. Feature extraction characterizes the parabolic shape by strengths of the diagonal and anti-diagonal slopes. At (x_s, y_s), a vector is extracted by performing a short-distance analysis along y on a y–z plane GPR image. The image is obtained at the cross-track location of x_s, and covers a window of four scans in y that consists of y_s, y_{s+1}, y_{y+2}, y_{s+3}. The resulting feature vector has 16 components. In such a way, feature vectors are extracted for every (x_s, y_s). The feature vectors are used to form observation sequences for computing confidence scores of mine and background. At the location (x_s, y_s), an observation sequence is formed by 15 adjacent feature vectors extracted at the locations of $(x_s, y_s - 7), \ldots, (x_s, y_s), \ldots, (x_s, y_s + 7)$. For details of the feature analysis procedure, refer to [10].

C. HMM and Baseline Model Training

The DHMM of mine has five states. The first and last states represent background that may surround a mine signature, and the second, third, and fourth states correspond to the positive slope, peak, and negative slope of a parabolic shape that characterizes mine signatures. The state transitions are constrained to be left-to-right [14], with the initial state probability specified as $\pi_1^{(1)} = \pi_2^{(1)} = 0.5$. The DHMM of background has three states. The state transitions are also constrained to be left-to-right, with the initial state probability specified as $\pi_1^{(2)} = 1.0$. Unlike the mine HMM, the states of background HMM have no clear physical meaning. In VQ, a large codebook is potentially more powerful for data representation, and a small codebook requires less computation and is more desirable in system implementation. To investigate such performance-computation tradeoffs, three different codebook sizes were experimented: 100, 50, and 25. For each specified codebook size, the self-organizing feature map (SOM) method [23] was first used to generate a codebook V. SOM is known to have the property of topology preservation in input-to-codeword mapping, and in the HMM-based landmine detection system [10], it was shown to be advantageous over conventional LBG-based VQ method. Once a codebook is generated, BW algorithm was then used to estimate the HMM parameters $\Lambda = (\lambda^{(1)}, \lambda^{(2)})$, where $\lambda^{(1)}$ was estimated from the mine data and $\lambda^{(2)}$ was estimated from the background data. As the result, three baseline models were generated, one for each codebook size.

D. Evaluation Procedure on Lane Data

The evaluation procedure on lane data basically follows that defined in [10] and is briefly summarized here. At each point (x, y) on a lane, the likelihood scores of the corresponding observation sequence are computed by the mine and background HMMs and the difference of the two scores, i.e.,

$$\text{score(mine)} - \text{score(background)}$$

is mapped to the range of 0 through 255. Binary confidence maps are produced by applying a set of thresholds to the difference values. For each threshold, the values that are above the threshold are set to 1, called 1-points, and the rest are set to 0. This thresholding step is in fact a likelihood ratio test, with the 1-points corresponding to testing outcomes of mine, and the 0-points testing outcomes of clutters. Noise cleaning is next applied to remove isolated 1-points. The connected 1-points form a detection region and each detection region is checked for a potential mine. A mine is detected if the centroid of a detection region falls within a square of 1 × 1 m^2 centered at the mine location; otherwise the detection region is designated as a false alarm. Detection rate is defined as the number of mines detected divided by the total number of mines in the lane. False-alarm rate is defined as the number of false alarms per square meter. For each lane, the mine detection results are summarized as the average detection rates at the quantized false-alarm rates of 0.01, 0.02, 0.03, 0.04, and 0.05, respectively, where the detection and false-alarm rates are averaged over different thresholds and different data-collection passes.

E. Discriminative Model Training

From each baseline model, MCE/GPD and EA training processes were carried out to generate multiple models, where MCE/GPD was used to generate multiple sets of HMM parameters Λ_i, and the EA was used to generate multiple codebooks V_i. Three ways of performing discriminative training were studied: MCE/GPD, EA, and EA followed by MCE/GPD. Following each type of training, a further step of model selection was carried out on subsets of data of arid site 1 and temperate site 2. The rational on the model selection step was that GPR data of different geographical sites often have very different characteristics due to soil type, weather condition, etc. Even though discriminatively trained models minimize classification errors on the training site (temperate site 1), they may not be optimal for the test sites (arid site 1 and temperate site 2), and GPR data within each site are better matched. Therefore, for a given test site, multiple models obtained from the training site can be first selected using a subset of data of the test site and the selected models can then be used for mine detection on the rest data of the test site. In the experiment, this select-and-test process was implemented by the existing data partition of each site, i.e., select by calibration set and test on blind test set, and vice versa. Specifically, on arid site 1 or temperate site 2, the calibration set was used to select five best models for mine detection on blind test set, and then the blind test set was used to select five best models for mine detection on the calibration set, and the procedure was carried out separately for models with different codebook sizes, i.e., 25, 50, 100. For practical purposes, it was desirable to use only a small amount of data for model selection, and this scenario was investigated in the following manners on arid site 1 and temperate site 2.

Case 1) For each test site, we randomly chose three lanes from the calibration set to select five best models for mine detection on the blind test set, repeating the procedure five times and averaging the detection and false-alarm results.

Case 2) For each test site, we randomly chose two lanes from the calibration set to select five best models for mine detection on the blind test set, repeating the procedure seven times and averaging the detection and false-alarm results.

Case 3) For each test site, we randomly chose a single lane from the calibration set to select five best models for mine detection on the blind test set, repeating the procedure 15 times and averaging the detection and false-alarm results.

In the above three cases, the training condition was limited to EA+MCE/GPD and the codebook size was fixed to be 25.

The metric for model selection was defined as the average detection rate over the false-alarm rate range of $0.02 \sim 0.04$. For model i, the detection rates were denoted as $d_{i,2}$, $d_{i,3}$, and $d_{i,4}$ at the false-alarm rates of 0.02, 0.03, and 0.04, respectively, and the average detection rate was defined as $s_i = 0.5 * d_{i,2} + 0.25 * d_{i,3} + 0.25 * d_{i,4}$, where the larger weight on $d_{i,2}$ emphasized the detection performance at the false-alarm rate of 0.02. The s_i's were sorted in an descending order and the top five models were selected.

1) MCE/GPD Training: The convergence behavior of MCE/GPD training was first studied. Classification errors versus training iterations is shown in Fig. 2, where the parameter γ of the sigmoid loss function was empirically chosen as 1.5, and the adaptation step size ϵ was taken as 0.4 and 0.1. It is observed that the training converged for both step sizes with slightly different convergence rates and final error counts. It is worth noting that in Fig. 2, the convergence behavior of MCE training is measured by classification error count rather than traditional mean-squared error of parameter estimates, where the former depends on more complicated factors of a classification system. As such, the observed lower final error count in using the step size 0.4 than 0.1 could be attributed to the ability of the sequential MCE training procedure to escape local convergence regions with a proper step size.

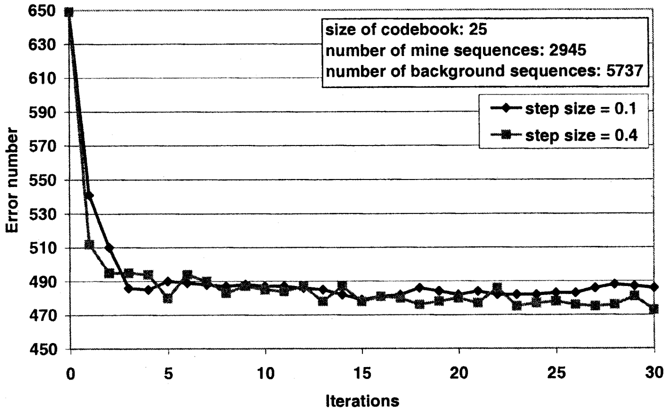


Fig. 2. Convergence behavior of MCE/GPD training of HMM parameters.

In general, in MCE training the step size parameter ϵ needs to be carefully chosen to balance learning rate and convergence behavior. A large ϵ leads to fast learning but may cause divergence, while a small ϵ leads to slow learning but is safe in convergence. Informal experimental evaluation revealed that the best step size ϵ was often data dependent, and it also depended on how well the baseline models fit the data. In the experiment, for each baseline model, the step-size ϵ was varied in the range of [0.1, 8], where in [0.1, 1] ϵ was increased by step of 0.1, and in [1, 8] ϵ was increased by step of 1. The parameterγ of the sigmoid loss function was fixed as 1.5, and the training iteration number was fixed as 30. Each step size ϵ_i led to an HMM Λ_i, and therefore multiple HMMs $\{\Lambda_i\}$ were produced from each baseline HMM Λ. MCE/GPD training was performed separately for the codebook sizes of 100, 50, and 25. Each combination of a codebook V and a HMM Λ_i comprised a model, and model selection was performed from the set $\{(V, \Lambda_i)\}$.

2) EA Training: From a baseline codebook V and an HMM Λ, the EP algorithm as described in Section II-B was used to generate a set of codebooks from the training dataset. The EP training parameters were empirically chosen as $G = 30$ and $\eta^{(0)} = 0.01$. For the codebook size of 25, the convergence behavior of EP is shown in Fig. 3 in terms of classification error versus training iteration, where the error count of an iteration was averaged from those of the 30 codebooks selected in the iteration. The error count was reduced with the iterations, and convergence was reached in about 50 iterations. On the platform of a 500-MHz Pentium III processor with a memory size of 256 MB, the FEP training time was evaluated to be 1 min.

EP training was performed separately for the codebook sizes of 100, 50, and 25, and the number of EP iterations was fixed to be 50 in each case. Each combination of a codebook V_i and the baseline HMM Λ comprised a model, and model selection was performed from the set $\{(V_i, \Lambda)\}$.

3) Combined MCE/GPD and EP Training: In the combined mode, for each given baseline model V and Λ, the EP algorithm

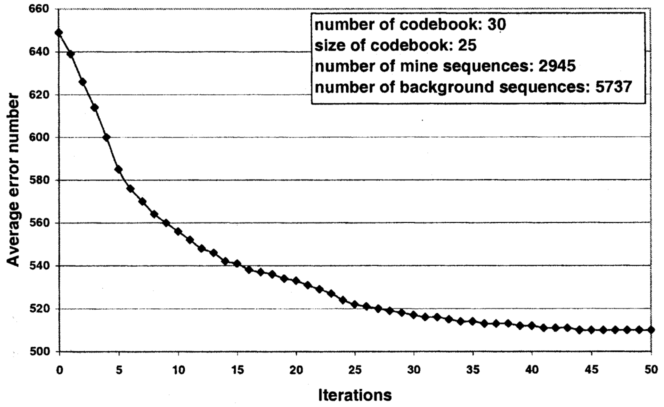


Fig. 3. Convergence behavior of EA training of codebook.

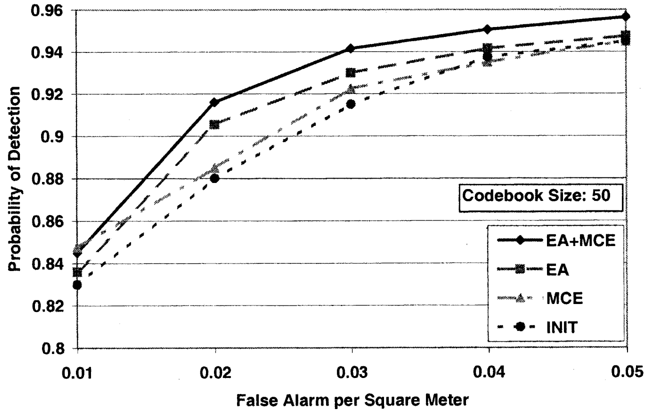


Fig. 4. Comparison of discriminative training methods for codebook size of 50.

was first used to generate 30 codebooks $V_1, V_2, \ldots, V_{30}$, and for each codebook V_i, the MCE/GPD algorithm was next employed to generate multiple sets of HMMs $\Lambda_{i,j}$ by varying the step size ϵ. Each combination of a codebook V_i and a HMM $\Lambda_{i,j}$ comprised a model, and model selection was performed from the set $\{V_i, \Lambda_{i,j}\}$.

It is worth noting that in the combined training mode, since EP improved model and hence reduced the space for further improvement, the experimental parameters of the subsequent MCE/GPD training might need adjustments. In this training mode, the training iteration number of MCE/GPD was reduced to ten, and on temperate site 2, the parameter γ was set as one, and ϵ was reduced by factors of 10^{-1} from the original range [0.1, 8].

F. Experimental Results

Experimental results are summarized in terms of training methods, codebook sizes, test sets, and data amount used in model selection. In each case, the results were averaged on those obtained from the blind test set (using models selected from the calibration set) and those obtained from the calibration set (using models selected from the blind test set).

In Fig. 4, the ROC curves for the training methods of MCE/GPD, EA, and EA+MCE/GPD are compared, where each curve represents average results over arid site 1 and temperate site 2, and the codebook size was fixed as 50. It is observed that the training method of EA+MCE/GPD produced best results. In particular, the detection rates achieved by the EA+MCE/GPD method at the false-alarm rate of 0.02 were about the same as the detection rates achieved by the baseline models at the false-alarm rate of 0.04, i.e., the false-alarm rate was approximately halved, indicating significant performance improvement. A performance trend similar to that of codebook size 50 was observed from ROC curves of codebook sizes 25 and 100, but details are omitted for the sake of space.

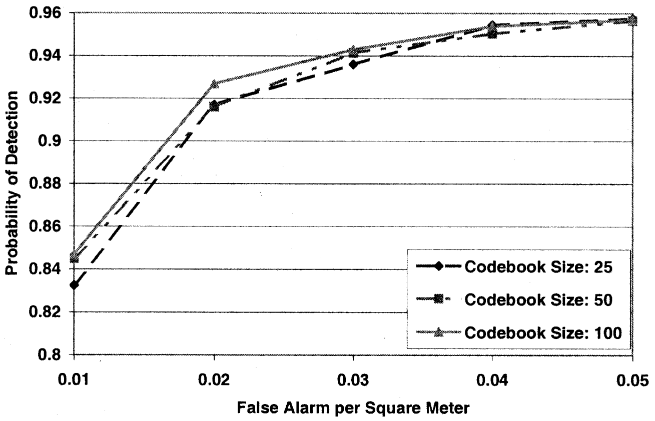


Fig. 5. Performance comparison on codebook sizes of 25, 50, and 100 for the training condition of EA+MCE.

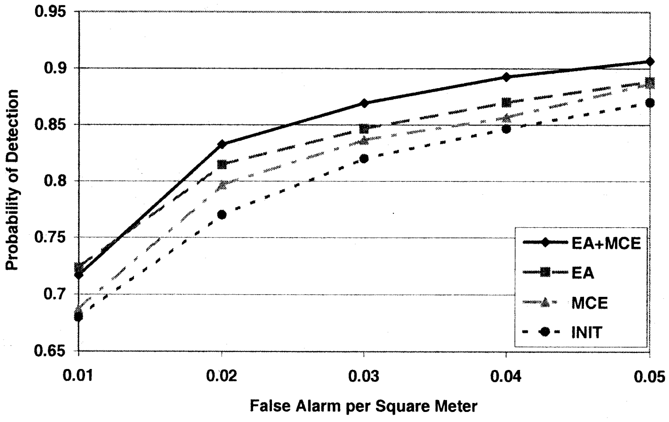


Fig. 6. ROC curves of calibration set of arid site 1. Model selection was performed by using blind test set of arid site 1.

In Fig. 5, the ROC curves corresponding to different codebook sizes are compared for the training condition of EA+MCE/GPD. The three codebooks are seen to have produced similar results, with the size-100 codebook being slightly better than the two smaller codebooks. Compared with the size-100 codebook, the size-25 codebook requires less computation in VQ and less memory space in codebook storage, and therefore it is an excellent choice in system design when computation power and memory space need to be considered.

In Figs. 6–9, the ROC curves of different training methods are shown separately for the four sets of test data: arid site 1 calibration, arid site 1 blind test, temperate site 2 calibration, and temperate site 2 blind test. Each curve was obtained by averaging the results of the three different codebook sizes of 100, 50, and 25. It is observed that the impact of EA+MCE/GPD

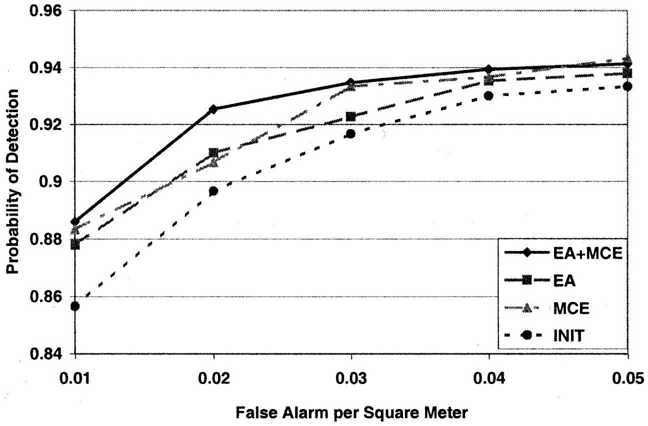


Fig. 7. ROC curves of blind test set of arid site 1. Model selection was performed by using calibration set of arid site 1.

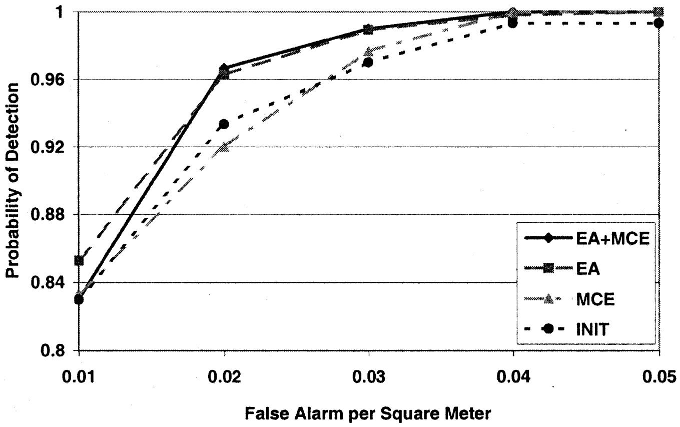


Fig. 8. ROC curves of calibration set of temperate site 2. Model selection was performed by using blind test set of temperate site 2.

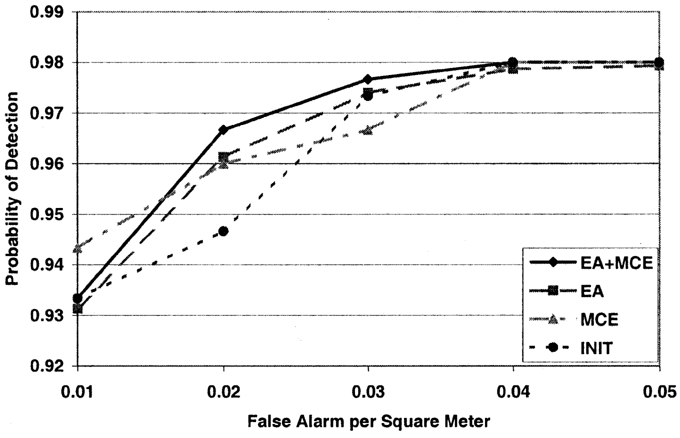


Fig. 9. ROC curves of blind test set of temperate site 2. Model selection was performed by using calibration set of temperate site 2.

on the mine detection performance was much more significant on the arid site 1 than the temperate site 2, and the EA alone training was more effective than the MCE/GPD alone training, in particular for temperate site 2.

In Figs. 10 and 11, the ROC curves for the three cases described in Section III-E are shown for the blind test sets of arid site 1 and temperate site 2, respectively. The results indicate that

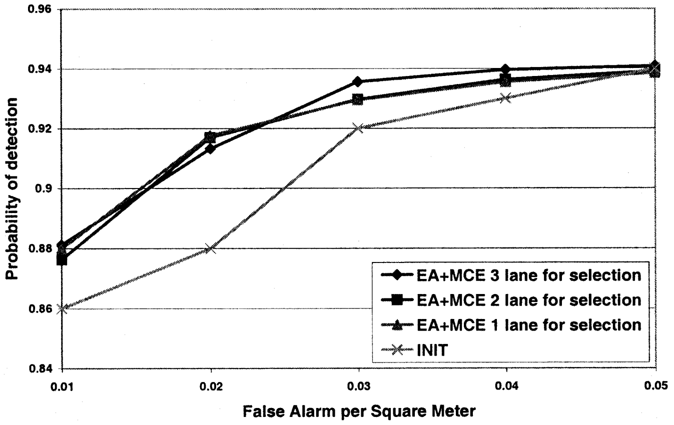


Fig. 10. ROC curves of blind test set of arid site 1. Model selection was performed by using 1, 2, and 3 calibration lanes of arid site 1, respectively.

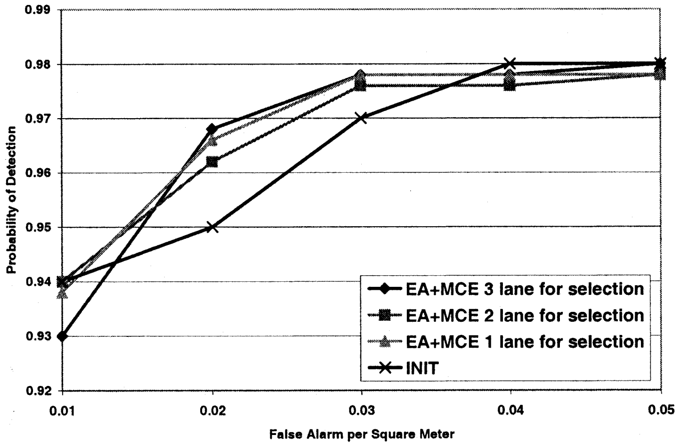


Fig. 11. ROC curves of blind test set of temperate site 2. Model selection was performed by using 1, 2, and 3 calibration lanes of temperate site 2, respectively.

model selection as performed by one through three lanes consistently improved the baseline results, and therefore it is feasible to use a small amount of lane data for model selection.

It can be observed that the evaluated competing methods are mainly distinguished by their mine detection rates in the operating regions of 0.02 ~ 0.03 false alarms per square meter. At the higher level of false alarms, i.e., 0.05 false alarms per square meter, the performance difference was small among these methods and geographical site became a dominant factor in the mine detection rate.

IV. Conclusion

In this work, minimum classification error training is proposed and evaluated for a DHMM-based landmine detection system. Three training scenarios were investigated, and it was found that codebook generation and selection by EA produced a larger impact on detection performance than DHMM parameter estimation by MCE/GPD, and the combined EA and MCE/GPD achieved the best overall system performance. At the same detection rate of the baseline DHMM system, the proposed EA+MCE/GPD training reduced false-alarm rate by a factor of two, indicating significant performance improvement. Except for the calibration set of arid site 1, the EA+MCE/GPD training exceeded the project target of above 90% mine detection at the false-alarm rate of 0.02 per square meter. Experimental results also indicate that the small codebook with 25 codewords was nearly as good as the large codebook with 100 codewords. For real system implementation, the small codebook is a good choice due to its lower requirement on computation power and memory space. In the current work, model selection needs to be performed by using a certain amount of lane data, and it is shown feasible by using a small amount of lane data. In the future work, mismatch compensation of training and test data characteristics may also be addressed from the perspective of model adaptation or transformation based on a small amount of lane data. Such techniques can be integrated with the proposed discriminative training methods to meet more stringent target of higher mine detection rate at lower level of false alarms.

Appendix

The gradient equation (1) is substantiated for the mine and background HMM parameters by the following set of equations:

$$\begin{aligned}\tilde{a}_{ij}^{(l)}(r+1) =& \tilde{a}_{ij}^{(l)}(r) - \frac{\epsilon}{r}\left\{\frac{\partial \mathcal{L}_1}{\partial \tilde{a}_{ij}^{(l)}}|_{\tilde{\Lambda}=\tilde{\Lambda}_r} 1(O_r \in C_1)\right.\\ &\left.+\frac{\partial \mathcal{L}_2}{\partial \tilde{a}_{ij}^{(l)}}|_{\tilde{\Lambda}=\tilde{\Lambda}_r} 1(O_r \in C_2)\right\}\\ \tilde{b}_{jk}^{(l)}(r+1) =& \tilde{b}_{ij}^{(l)}(r) - \frac{\epsilon}{r}\left\{\frac{\partial \mathcal{L}_1}{\partial \tilde{b}_{jk}^{(l)}}|_{\tilde{\Lambda}=\tilde{\Lambda}_r} 1(O_r \in C_1)\right.\\ &\left.+\frac{\partial \mathcal{L}_2}{\partial \tilde{b}_{jk}^{(l)}}|_{\tilde{\Lambda}=\tilde{\Lambda}_r} 1(O_r \in C_2)\right\}\end{aligned}$$

with $l = 1, 2$, $i, j = 1, 2, \ldots, N$, $k = 1, 2, \ldots, M$. Define the indicator functions

$$\delta(u = i, v = j) = \begin{cases} 1, & \text{if } u = i, v = j \\ 0, & \text{otherwise} \end{cases}$$

$$\delta(u = i) = \begin{cases} 1, & \text{if } u = i \\ 0, & \text{otherwise.} \end{cases}$$

The partial derivatives in the above equations are derived as

$$\begin{aligned}\frac{\partial \mathcal{L}_m}{\partial \tilde{a}_{ij}^{(l)}} =& (-1)^{1+m-l}\gamma \mathcal{L}_m(1-\mathcal{L}_m)\\ &\cdot\left\{\sum_{t=1}^{T}\delta(q_{t-1}=i, q_t=j) - a_{ij}^{(l)}\sum_{t=1}^{T}\delta(q_{t-1}=i)\right\}\\ \frac{\partial \mathcal{L}_m}{\partial \tilde{b}_{jk}^{(l)}} =& (-1)^{1+m-l}\gamma \mathcal{L}_m(1-\mathcal{L}_m)\\ &\cdot\left\{\sum_{t=1}^{T}\delta\left(q_t=j, Q_V(o_t)=k\right) - b_{jk}^{(l)}\sum_{t=1}^{T}\delta(q_t=j)\right\}\end{aligned}$$

where $l = 1, 2$, $m = 1, 2$, and q_t's are determined by Viterbi algorithm. The initial state probabilities π_i are fixed to the specified values and not estimated.

References

[1] "Hidden killers: The global landmine crisis," U.S. Dept. State Rep., Washington, DC, Pub. 10575, Sept. 1998.

[2] T. T. Witten, "Present state of the art in ground-penetrating radars for mine detection," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets III*, Orlando, FL, 1998, pp. 576–586.

[3] M. D. Patz and M. A. Belkerdid, "Evaluation of a model-based inversion algorithm for GPR signal processing with correlation for target classification," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets III*, Orlando, FL, 1998, pp. 598–604.

[4] H. T. Kaskett and J. T. Broach, "Automatic mine detection algorithm using ground penetrating radar signatures," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 942–952.

[5] S.-H. Yu, R. K. Mehra, and T. R. Witten, "Automatic mine detection based on ground penetrating radar," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 961–972.

[6] D. Careevic, "Clutter reduction and target detection in ground penetrating radar data using wavelet," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 973–978.

[7] X. Xu, E. L. Miller, and C. M. Rappaport, "Statistically-based sequential detection of buried mines from array ground penetrating radar data," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 1063–1074.

[8] A. van der Merwe, I. J. Gupta, and L. Peters Jr., "A clutter reduction technique for GPR data from mine like targets," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 1094–1105.

[9] P. D. Gader, B. Nelson, H. Frigui, G. Vaillette, and J. Keller, "Landmine detection in ground penetrating radar using fuzzy logic," *Signal Process.*, vol. 80, no. 6, pp. 1069–1084, June 2000.

[10] P. Gader, M. Mystkowski, and Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Trans. Geosci. Remote sensing*, vol. 39, pp. 1231–1244, June 2001.

[11] J. M. Stiles, P. Parra-Bocaranda, and A. Apte, "Detection of object symmetry using bistatic and polarimetric GPR observations," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets IV*, Orlando, FL, 1999, pp. 992–1002.

[12] S. L. Tantum, Y. Wei, V. S. Munshi, and L. M. Collins, "A comparison of algorithms for landmine detection and discrimination using ground penetrating radar," in *Proc. SPIE Conf. Detection and Remediation Technologies for Mines and Minelike Targets VII*, Orlando, FL, 2002, pp. 728–735.

[13] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. ASSP-28, pp. 84–95, Apr. 1980.

[14] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[15] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, pp. 257–266, May 1997.

[16] 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, pp. 2345–2372, Nov. 1998.

[17] T. Back and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 1–23, 1993.

[18] M. G. Rahim, B.-H. Juang, and C.-H. Lee, "Discriminative utterance verification for connected digit recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, pp. 266–277, May 1997.

[19] X. Yao, "Global optimization by evolutionary algorithms," in *Proc. Parallel algorithms/Architecture Synthesis, 2nd Aizu Int. Symp.*, Mar. 1997, pp. 282–291.

[20] T. Rudolph, "Discriminative codebook design for critical word recognition using evolution strategies," in *Proc. IEEE Int. Conf. Evolutionary Computation*, May 1996, pp. 67–70.

[21] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 82–102, July 1999.

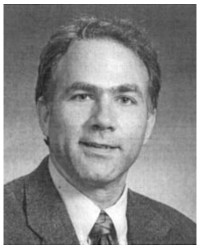
[22] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Networks*, vol. 5, pp. 3–14, Jan. 1994.

[23] T. Kohonen, "The self-organizing feature map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.

Yunxin Zhao (S'86–M'88–SM'94) received the Ph.D degree from University of Washington, Seattle, in 1988.

She is currently a Professor in the Department of Computer Engineering and Computer Science, University of Missouri, Columbia. She was a Senior Research Staff and Project Leader of the Speech Technology Laboratory, Panasonic Technologies, Inc., Santa Barbara, CA, from 1988 to 1994. She was an Assistant Professor in the Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign from 1994 to 1998. Her research interests are in spoken language processing, automatic speech recognition, multimedia interface, multimodal human–computer interaction, statistical pattern recognition, statistical blind systems identification and estimation, speech and signal processing, and biomedical applications.

Dr. Zhao was an Associate Editor of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING and a Member of IEEE Speech Technical Committee. She received the NSF Career Award in 1995 and is listed in *American Men and Women of Science,* February 1998.



Paul Gader (M'87–SM'99) received the Ph.D. degree in mathematics for parallel image processing and applied-mathematics-related research from the University of Florida, Gainesville, in 1986.

He has worked as a Senior Research Scientist at Honeywell's Systems and Research Center, Minneapolis, MN, as a Research Engineer and Manager at the Environmental Research Institute of Michigan (ERIM), Ann Arbor, and as a faculty member at the University of Wisconsin, Oshkosh, the University of Missouri, Columbia, and the University of Florida, where he is currently employed. He performed his first research in image processing in 1984 when he worked on algorithms for detection of bridges in Forward Looking Infra-Red (FLIR) imagery. At ERIM and later at the University of Missouri, he led teams involved in the research and development of real-time handwritten-address recognition systems for the U.S. Postal Service. He developed, implemented, and tested image processing, neural network, and fuzzy-set-based algorithms for handwritten-digit recognition and segmentation, numeric field recognition, word recognition and segmentation, and line segmentation. He has also worked on a number of other imaging projects, including medical imaging, vehicle detection and recognition, and biomedical pattern recognition. He has been active in land mine detection algorithm research since 1996. He has led teams that devised and field tested several real-time algorithms for mine detection. He has served as Technical Director of the University of Missouri Multidisciplinary University Research Initiative (MURI) on Humanitarian Demining for two years. He is currently involved in several landmine detection projects, including hand-held and ground-based mine detection systems, acoustic mine detection, detection of trip-wires, and is a member of a new MURI team investigating the science of land target spectral signatures. He has published over 150 technical publications in the areas of image and signal processing, applied mathematics, and pattern recognition, including over 45 refereed journal articles.

Ping Chen received the B.S. degree in electrical engineering from NanJing University, Nanjing, China, in 1998, and the M.S. degree in computer engineering and computer science from University of Missouri, Columbia (UMC), in 2002.

She is currently with REI Systems, Inc., Vienna, VA. She had been with UMC's Computational Intelligent Lab from 2000 to 2002. Her research interest was in the area of pattern recognition, signal processing, and statistic modeling.

Yue Zhang received the M.S. degree in computer science from University of Missouri, Columbia, in 2000.

He is currently working on the networked information system at Conexant System Inc., Newbury Park, CA.