Dear Author,

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

**Articles in Special Issues**: Please ensure that the words ‘this issue’ are added (in the list and text) to any references to other articles in this Special Issue.

<table>
<thead>
<tr>
<th>Location in article</th>
<th>Query / remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Please update Refs. [30,96].</td>
</tr>
</tbody>
</table>

**Uncited references**: References that occur in the reference list but not in the text – please position each reference in the text or delete it from the list.

**Missing references**: References listed below were noted in the text but are missing from the reference list – please make the list complete or remove the references from the text.

Electronic file usage

Sometimes we are unable to process the electronic file of your article and/or artwork. If this is the case, we have proceeded by:

- [ ] Scanning (parts of) your article
- [ ] Rekeying (parts of) your article
- [ ] Scanning the artwork

Thank you for your assistance.
An evaluation of several fusion algorithms for anti-tank landmine detection and discrimination

Hichem Frigui a,*, Lijun Zhang a, Paul Gader b, Joseph N. Wilson b, K.C. Ho c, Andres Mendez-Vazquez b

a CECS Dept., University of Louisville, Louisville, KY, USA
b CISE Dept., University of Florida, Gainesville, FL, USA
c ECE Dept., University of Missouri, Columbia, MO, USA

A R T I C L E   I N F O

Article history:
Received 16 January 2008
Received in revised form 1 October 2008
Accepted 30 October 2009
Available online xxxx

Keywords:
Landmine detection
Bayesian fusion
Context-dependent fusion
Dempster–Shafer
Borda count
Decision template
Fuzzy integral

A B S T R A C T

Many algorithms have been proposed for detecting anti-tank landmines and discriminating between mines and clutter objects using data generated by a ground penetrating radar (GPR) sensor. Our extensive testing of some of these algorithms has indicated that their performances are strongly dependent upon a variety of factors that are correlated with geographical and environmental conditions. It is typically the case that one algorithm may perform well in one setting and not so well in another. Thus, fusion methods that take advantage of the stronger algorithms for a given setting without suffering from the effects of weaker algorithms in the same setting are needed to improve the robustness of the detection system. In this paper, we discuss; test; and compare seven different fusion methods: Bayesian, distance-based, Dempster–Shafer, Borda count, decision template, Choquet integral, and context-dependent fusion. We present the results of a cross validation experiment that uses a diverse data set together with results of eight detection and discrimination algorithms. These algorithms are the top ranked algorithms after extensive testing. The data set was acquired from multiple collections from four outdoor sites at different locations using the NIITEK GPR system. This collection covers over 41,807 m² of ground and includes 1593 anti-tank mine encounters.

© 2009 Published by Elsevier B.V.

1. Introduction

It is estimated that over 100 million landmines are buried in over 80 countries and that 26,000 people a year are killed or maimed by a landmine [1]. Detection and removal of landmines is a significant research problem [2–5]. The research problem for data analysis is to determine how reliably landmines can be detected and distinguished from other subterranean objects using sensor data. Difficulties arise from the variability of landmine types, soil and weather conditions, terrains, and so on. Traditional fielded approaches use metal detectors. Unfortunately, many landmines contain little metal. Ground penetrating radar (GPR) offers the promise of detecting landmines with little metal. Although several approaches to detecting landmines and discriminating landmines from clutter using GPR have been investigated [6–15], acceptable results have been elusive [16–18]. Although systems often achieve high detection rates, it is difficult to achieve the required low false alarm rates. Moreover, algorithm performance can vary significantly. Therefore, fusion methods that take advantage of the strengths of individual algorithms, overcome their weaknesses, and achieve a higher accuracy than any individual algorithm are needed.

Multi-classifier, multi-algorithm, and multi-sensor fusion are critical components in landmine detection. Buried objects interact with the soil and any potential covering of the soil (such as a road surface). Physical properties of soil can vary significantly in small areas. For example, soil can be a heterogeneous mixture of soil types layered with a thin layer of top soil covering clay or asphalt covering gravel covering soil. Soil can have significantly varying density in a small region [19]. Roots of vegetation hold water. Rain or snow lead to variable moisture in the soil. Minerals can significantly affect the radar propagation through soil. In addition, the mine case can interact with different soils in different ways. For example, plastic casings have very similar electrical properties as soils under some conditions. Wood casings can absorb moisture. All these factors can have significant effects on GPR data and are generally unknown to an autonomous algorithm due to the wide variability over a small range. The implication for autonomous detection is that different types of algorithms are useful for different conditions. These different algorithms must use different signal conditioning, or Preprocessing, and feature extraction.

The objective of this paper is to present results of evaluating eight different anti-tank landmine detection and discrimination algorithms and
the fusion of these algorithms using seven different methods. The
85 generality, computational cost, and interpretability of the fusion
86 methods is analyzed using a cross validation experiment that uses
87 a diverse data set acquired from four outdoor test sites at different
88 geographic locations. This collection covers over 41,807 m² of
ground and includes 1393 anti-tank mine encounters. This collection
89 contains multiple sub-collections taken at different times of
90 the year and at very different locations in the United States as well
91 as in Europe. Therefore, the experimental results, although not
92 completely independent of mine type, soil conditions, etc., are
93 probably at least as independent as any published results.
94 Section 2 describes the GPR data, Preprocessing, and prescreen-
95 ing. Section 3 outlines the distinct anti-tank landmine discrimina-
96 tion algorithms. Section 4 discusses the seven methods for fusing
discrimination algorithms. Experimental results and analyses are
97 presented in Section 5. Section 6 concludes.

2. Data preprocessing and prescreening

In this section, we briefly describe the GPR data, Preprocessing
steps, and Prescreening. More detailed descriptions are in [20,21].

2.1. GPR data

The input data consist of a sequence of raw GPR measurements
99 collected by a vehicle-mounted GPR array [22] (see Fig. 1a). The
100 GPR collects 24 channels of data. Adjacent channels are spaced
101 approximately 5 cm apart in the cross-track direction, and se-
102 quences (or scans) are taken at approximately 5 cm down-track
103 intervals. The system uses an antenna that generates a wide-band
104 pulse from 200 MHz to 7 GHz. Each A-scan, that is, the measured
105 waveform collected in one channel at one down-track position,
106 contains 416 GPR time samples, each corresponding to roughly
107 16 ps. We often refer to the time index as depth although, since
108 the radar wave travels through different media, this index does
109 not represent a uniform sampling of depth. Thus, we model GPR
110 input data as a three-dimensional matrix of sample values,
111 \( x(z, x, y), z = 1, \ldots, 416; x = 1, \ldots, 24; y = 1, \ldots, N_y \), where
112 \( N_y \) is the total number of collected scans, and the indices \( z, x, \) and \( y \) repre-
113 sent depth, cross-track position, and down-track positions re-
114 spectively. GPR input data is illustrated in Fig. 1b.

Fig. 2 displays down-track B-scans (sequences of A-scans from a
115 single channel) and cross-track B-scans (sequences of A-scans from
116 a single scan). The surveyed object position is highlighted in each
117 figure.

2.2. Data preprocessing

Preprocessing is an important step to enhance the mine signa-
tures. The algorithm first identifies the ground bounce location as
the global maximum of the signal and aligns the A-scans using
these maxima. This alignment is necessary because the system
cannot maintain the radar antenna at a fixed distance above the
ground. The early time samples of each signal, up to few samples
beyond the ground bounce are discarded. The remaining samples
are divided into \( N \) depth bins which will be processed indepen-
dently. The reason for this segmentation is to compensate for the
high contrast between responses from deeply buried and shallow
anomalies.

2.3. Anomaly detection

Our algorithm applies a Prescreener to reduce the volume of
GPR data to be inspected. The Prescreener identifies distinct alarm
locations in the data. It was designed to provide a high probability
of detection so that more computationally intensive discrimination
processing can be performed. False alarms are alarms that do not
 correspond to mines. The objective of the feature-based detection
algorithms and their fusion is to distinguish between Prescreener
alarms corresponding to landmines from false alarms. We use
the Duke University NUKEv6 Prescreener, a variant of the least
mean squares (LMS) Prescreener [20]. A version of this Prescreener
was implemented in real-time in the system in Fig. 1. This Prescre-
ener is applied to the energy at each depth bin, and assigns a con-
fidence value to each point in the cross-track, down-track plane
based on its contrast with a neighboring region. The cross-track
\( x_s \) and down-track \( y \) positions of the centers of algorithmically
determined mine-like components are reported as alarm positions
for further processing.

3. Discrimination algorithms

Generally, automated landmine discrimination algorithms
consist of three phases: Preprocessing, feature extraction, and con-
fidence assignment. Preprocessing performs tasks such as normal-
izing data, correcting for variations in height and speed, and
removing stationary effects due to the system response. Previous
methods include wavelets and Kalman filters [23,24], subspace
methods and polynomial matching [25], and subtracting optimally
shifted and scaled reference vectors [26]. Feature extraction
reduces the Preprocessed data to a lower-dimensional, salient set
of values that represent the data. The principal component

---

Please cite this article in press as: H. Frigui et al., An evaluation of several fusion algorithms for anti-tank landmine detection and discrimination, Infomat.
Fusion (2009), doi:10.1016/j.inffus.2009.10.001

---

Fig. 1. GPR data collection: (a) NIITEK vehicle-mounted GPR system; and (b) an example of GPR scans.
transform is a common feature extraction tool \cite{27}, as are wavelets \cite{23}, image processing based differentiation \cite{6}, and Hough and Radon transforms \cite{4}. Confidence assignment can be performed using methods such as Bayesian \cite{4}, hidden Markov Models \cite{6,28}, fuzzy logic \cite{5}, rules and order statistics \cite{21}, neural networks, or nearest neighbor classifiers \cite{7}.

Here we consider seven specific algorithms of distinct character. These algorithms have performed well in extensive field testing, and are being considered for real-time implementation in handheld and vehicle-mounted GPR systems. These algorithms are highlighted in the following sections.

3.1. HMM detector

The HMM algorithm \cite{6,28} treats the down-track dimension as the time variable and produces a mine confidence at positions, \((x,y)\), on the surface being traversed. A sequence of observation vectors is produced for each down-track point and depth. These observation vectors encode the degree to which edges occur in the diagonal and anti-diagonal directions. In particular, for every point \((x_s,y_s)\), the strengths for the positive/negative diagonal/anti-diagonal edges is computed. The observation vector at a point \((x_s,y_s)\) consists of a set of four features that encode the maximum edge magnitude over multiple depth values around \((x_s,y_s)\). The HMM algorithm has a background and a mine model. Each model produces a probability. The probability produced by the mine (background) model is an estimate of the probability of the observation sequence given that there is a mine (background) present. The log of the ratio of the probabilities is the confidence.

3.2. EHD detector

This algorithm uses translation invariant features based on the edge histogram descriptor (EHD) of the 3-D GPR signatures, and a possibilistic K-Nearest Neighbors (K-NN) rule for confidence assignment \cite{29}. The EHD captures the signature’s texture. Specifically, each 3-D signature is divided into sub-signatures, and the local edge distribution for each sub-signature is represented by a histogram. To generate the histogram, local edges are categorized into five types: vertical, horizontal, diagonal (45° rising), anti-diagonal (45° falling), and non-edges. A set of alarms with known ground truth is used to train the decision-making process. These labeled alarms are clustered to identify a small number of representatives that capture signature variations due to differing environmental conditions and mine types, etc.

3.3. SPECT detector

This detector aims at capturing the characteristics of a target in the frequency domain using the energy density spectrum (EDS). It extracts the spectral correlation feature (SCF) which is computed using similarity to mine prototypes \cite{30}. The EDS is estimated using three main steps: Preprocessing, whitening, and averaging. After alignment, Preprocessing removes the data above and near the ground surface to avoid an EDS that is dominated by the ground response. The whitening step equalizes the background spectrum so the estimated EDS reflects the actual spectral characteristics of an alarm. Averaging reduces the variance in the EDS.

![Fig. 2. NIITEK Radar down-track and cross-track B-scans pairs for three alarms.](image-url)
3.4. GEOM detector

This algorithm computes geometric features in multiple, whitened depth bins which are two-dimensional images with cross-track and down-track axes. The features are inputs to a Feed-forward Ordered-Weighted-Average (FOWA) network [31] that is trained to maximize the area under the Receiver Operating Characteristic (ROC) curve [32]. The features used are compactness, eccentricity, solidity, and area to filled area ratio. These features are based on the observation that the whitened energy for mines often has a compact, solid, and circular shape whereas non-mine-like objects produce an irregular shape.

3.5. TFCM detector

The Texture Feature Classification Method (TFCM) detector [33] is a three-dimensional extension of the algorithm by Horng [34]. The algorithm transforms a block of GPR data into a block of integer codes. The code at each point in a block is generated by considering several differences in GPR intensity values over a 3 × 3 × 3 window centered at the point. The differences are thresholded producing a string of zeros and ones, which are then mapped to the integer codes, the details of which are described in the references. Statistical textures features, such as entropy, variance, and co-occurrence, are then computed on the blocks of codes and transformed into feature vectors. Relevance Vector Machines (RVMs) use the features to produce a confidence that an alarm represents a landmine.

3.6. GMRF detector

The Gaussian-Markov Random Field (GMRF) detector [35] is based on a transmission line model of the time-domain GPR response to the subsurface. The model represents the GPR as a sequence of dielectric discontinuities. Each discontinuity is parameterized by a location and a gain parameter. These parameters are characterized statistically using a Gaussian-Markov Random Field. A generalized likelihood ratio test is then used to assign a confidence that an alarm represents an anti-tank landmine.

3.7. GFT detector

The Gaussian Fit (GFT) detector [36] calculates the parameters of a Gaussian pulse which best fits the spatial energy distribution of target responses to GPR. The output features are the goodness of fit, the pulse width, and pulse gain. More specifically, the spatial shape of the summed energy from a cross-track scan is compared to the shape of a Gaussian pulse. If \( x \) represents position in down-track scans, and \( E \) represents the energy, we find the \( \sigma, \mu, \alpha \) to minimize the root mean square error (RMSE) between \( E(x) \) and \( \chi(x) = \mu + \exp(\alpha(-x_0-x)/\sigma^2) \). The output features are then \( \sum_{x} (E(x) - f(x)), \sigma, \mu, \) and \( \alpha \).

The above discrimination algorithms were developed by researchers at the Universities of Missouri, Louisville, Florida, as well as Duke University. They are independently developed and have many differences in GPR Preprocessing and normalization, feature extraction, and classification methodologies. Since the descriptions of almost all of these algorithms are contained in detail in the references, and take many pages to describe in detail, they cannot be described in detail here. However, in feature extraction alone one can see many differences. The anomaly detector simply looks for locations that are different from the background. It uses masks oriented in the C-scan direction. The HMM detector looks at variable length sequences of edges. The EHD detector looks at fixed length representations of edges. All three previous algorithms used the down-track and cross-track time-domain GPR.

The SPECT detector looks at features in the frequency domain. The GEOM detector calculates feature based on geometric shape in C-scans. The TFCM detector looks for texture features in three-dimensional blocks of time domain data, GMRF, and the GFT detector looks at energy in the cross-track direction. Thus, in the feature extraction process alone, one can see that these algorithms vary widely in the focus and processing.

Despite all of the above differences, one cannot assume that these algorithms are statistically independent. In fact, we know that some of them could be highly correlated. For instance, both the EHD and the HMM detectors could assign low confidence values to alarms with weak edges. The fusion algorithms that we are considering (described in the next section) address the independence issue to various degrees. For instance, the Bayes fusion and the Mahalanobis distance fusion do not make the independence assumption and use full covariance matrices to normalize and decorrelate the detectors outputs. Similarly, the Choquet integral considers all possible subsets of detectors and promotes sparsity. Thus, it will tend to identify the smallest subset of uncorrelated detectors. Other fusion methods do not consider the detectors dependency at all. One of the goals of this experiment is to compare these fusion methods with respect to this dependency issue.

4. Combination of multiple classifiers

4.1. Background

For complex detection and classification problems involving data with large intra-class variations and noisy inputs, perfect solutions are difficult to achieve, and no single source of information can provide a satisfactory solution. As a result, combination of multiple classifiers (or multiple experts) is playing an increasing role in solving these complex pattern recognition problems, and has proven to be viable alternative to using a single classifier. Classifier combination is mostly heuristic and is based on the idea that classifiers with different methodologies or different features can have complementary information. Thus, if these classifiers cooperate, group decisions should be able to take advantages of the strengths of the individual classifiers, overcome their weaknesses, and achieve a higher accuracy than any individual’s.

Methods for combining multiple classifiers can be classified into two main categories: classifier selection and classifier fusion. Classifier selection methods assume that the classifiers are complementary, and that their expertise varies according to the different areas of the feature space. For a given test sample, these methods attempt to predict which classifiers are more likely to be correct. Some of these methods consider the output of only one classifier to make the final decision [37]. Others, combine the output of multiple “local expert” classifiers [38]. Classifier fusion methods assume that the classifiers are competitive and are equally experienced over the entire feature space. For a given test sample, the individual classifiers are applied in parallel, and their outputs are combined in some manner to take a group decision.

Over the past few years, a variety of schemes have been proposed for combining multiple classifiers. The most representative approaches include majority vote [39], Borda count [40], average [41], weighted average [42], Bayesian [43], and probabilistic [44]. Most of the above approaches assume that the classifier decisions are independent. However, in practice, the outputs of multiple classifiers are usually highly correlated. Therefore, in addition to assigning fusion weights to the individual classifiers, it is desirable to assign weights to subsets of classifiers to take into account the interaction between them. Fusion methods based on the fuzzy integral [45,46] and Dempster–Shafer theory [47,48] have this desirable property.
Another way to categorize classifier combination methods is based on the way they select or assign weights to the individual classifiers. Some methods are global and assign a degree of worthiness, that is averaged over the entire training data, to each classifier. Other methods are local and adapt the classifiers’ worthiness to different data subspaces. Intuitively, the use of data-dependent weights, when learned properly, provides higher classification accuracy. This approach requires partitioning the input samples into regions during the training phase. The partition can be defined from the space of individual classifier decisions [49], according to which classifiers agree with each other [40], or by features of the input space [50,51]. Then, the best classifier for each region is identified and is designated as the expert for this region [52]. Conversely, the partitioning can be defined such that each classifier is an expert in one region [37]. This approach may be more efficient, however, its implementation is not trivial. In the classification phase, the region of an unknown sample is identified, and the output of the classifier responsible for this region is used to make the final decision. Data partition and classifier selection could also be made dynamic during the testing phase [53,54]. In this case, the accuracy of each classifier (with respect to the training samples) is estimated in local regions of the feature space in the vicinity of the test sample. The most accurate classifier is selected to classify the test sample.

### 4.2. Notation

Let \( D_1, D_2, \ldots, D_l \) denote the \( L \) algorithms to be fused, and let \( w_1, w_2, \ldots, w_l \) denote the \( C \) classes. Each algorithm, \( D_i \), extracts a set of features, \( F_i \), and assigns a confidence value \( y_i \) to each of the \( C \) classes. In the proposed landmine application, we have \( L = 8 \), where \( D_1, D_2, \ldots, D_8 \) correspond to the prescreener (NUKEv6), EHD, HMM, Spect, Geom, TFCM, GHT, and GMRF algorithms respectively. Additionally, we have \( C = 2 \) where \( w_1 \) denotes the mine class and \( w_2 \) denotes the clutter class. We note that the prescreener is not a feature-based algorithm, and thus, it does not generate a set of features (i.e., no \( F_i \)).

### 4.3. Bayesian-based fusion

Bayesian data fusion [55] is based on Bayesian decision theory which is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the trade-offs between various classification decisions using probability and the costs that accompany such decisions, Bayesian data fusion has been studied extensively in the literature (e.g., [55–57]). This approach has the advantage of being able to incorporate a priori knowledge about the likelihood of the hypothesis being tested, and when empirical data are not available, it is possible to use subjective estimates of the prior probabilities. Moreover, from a statistical point of view, the use of Bayes rule should provide the optimal decision. Unfortunately, the proper use of Bayes requires the joint probability density functions to be known. This information is usually not available and may be difficult to estimate from the data. Other disadvantages of the Bayesian approach include complexities when dealing with multiple potential hypotheses and multiple conditionally dependent events, and the inability to account for general uncertainty [56,57]. Thus, Bayesian data fusion is best suited to applications where prior parameters are available, there is no need to represent ignorance, and where conditional dependency can be easily modeled through probabilistic representation.

Bayesian fusion has been applied to target identification [58], image analysis [59], and many other applications [55]. It has also been applied to the problem of anti-personnel landmine detection [60,61], and the results were compared to other fusion methods. In [60], only synthetic data were used, and in [61] a very small data set was used. Thus, the results were not conclusive.

Let \( v \) represents the output of all \( L \) algorithms to be fused, i.e., \( v = [y_1, y_2, \ldots, y_L] \). Within the Bayesian framework, \( v \) is considered a random variable with a distribution that depends on the state of nature. Using Bayes formula, we first compute the posterior probability using

\[
p(w_i|v) = \frac{p(v|w_i)p(w_i)}{p(v)}
\]

(1)

Then, \( v \) is assigned to the class with maximum posterior probability, i.e.,

\[
v \in w_j \text{ if } p(w_j|v) = \max_{i=1, 2} p(w_i|v).
\]

(2)

In (1), \( p(w_i) \) is the prior probability of class \( i \) and \( p(v|w_i) \) is the class conditional density. The prior \( p(w_i) \) is usually provided by an expert, or estimated using the relative proportions of training data from each class. Similarly, \( p(v|w_i) \) can be estimated from the training data.

Our data consist of multiple subsets of mines/clutter signatures collected with the same hardware at different times and under different conditions. Moreover, many mines are of the same type and/or buried at the same depth. Thus, it is reasonable to assume that the detectors will assign confidence values that are consistent with these conditions, and that the confidence values of all detectors tend to form clusters in the confidence space. Consequently, we model \( p(v|w_i) \) by a mixture of \( M \) Gaussian distributions, i.e.,

\[
p(v|w_i) = \sum_{k=1}^{M} p(v|w_k|w_i)p(w_k),
\]

(3)

where each \( p(v|w_k|w_i) \) is a multi-variate Gaussian. In general, we have \( p(v|w_k) \gg p(w_i) \). However, the risk associated with missing a mine is much higher than the risk associated with detecting a false alarm. Since we cannot quantify the risks, and the priors can change from one site to another and depend on the settings of the prescreener, we simply assume that these two factors cancel each other, and let \( p(w_i) = p(w_2) \).

In our experiments, we let \( v \) include the output of the seven detection algorithms and the prescreener, i.e., \( v = [y_1, y_2, \ldots, y_7] \).

The means \( \mu_k \), covariance matrices \( \Sigma_k \), number of components \( M \), and the mixing coefficients \( p(w_k) \) for the \( M \) components of class \( i \) are learned from the training data using the competitive agglomeration clustering algorithm [62]. Instead of using (2) to label the test data, we assign a soft confidence value using

\[
Conf_{i,k} = p(w_i|v).
\]

(4)

### 4.4. Mahalanobis distance-based fusion

The Mahalanobis distance-based approach (MD) is a variation of the Bayesian approach [63]. It models the distribution of \( v \) in each class \( i \), by a multi-variate Gaussian and therefore represents the eccentricity of the mine and clutter distributions. The Mahalanobis distances to the mine and clutter classes of a test alarm \( v \) are computed by

\[
D_k = (v - \mu_k)^T \Sigma_k^{-1} (v - \mu_k),
\]

(5)

where \( X = M \) and \( X = C \) denote the mine and clutter classes, respectively. The fusion confidence is the weighted difference between the distances:

\[
Conf_{MD} = D_m + xD_c.
\]

(6)

The value \( x \) in (6) provides a means of controlling the contribution of the distance to the clutter class to the fusion confidence. It is
computed from the training data to minimize the average false
alarm rate over the range of probability of detection from 92% to
96% [63]. This range was chosen since our long term goal is in
probabilities of detection around 95% and this interval contains
that range. Based on our experience the mines with confidence
so low that they are within the last 4% of the mines detected tend
to be lucky detects, i.e. they do not really produce useful signatures
and therefore should not be included in the optimization. This is
why the range is not symmetric around 95%.

The use of Mahalanobis distance has the advantage of normal-
izing the features and removing their correlation before fusing
them. This is reflected by the use of the covariance matrix in the
distances (5). Furthermore, the generation of confidences using
(6) is based on the theoretically sound likelihood ratio when \( \nu \) is
assumed to be Gaussian and when \( \alpha = 1 \) [64].

4.5. Dempster–Shafer based Fusion

Dempster–Shafer (DS) is a mathematical theory of evidence for
representing uncertain knowledge [65,66]. In a finite discrete
space, DS can be interpreted as a generalization of probability the-
ory where probabilities are assigned to sets as opposed to mutually
exclusive singletons. In DS, evidence can be associated with multi-
ple possible events, e.g., sets of events. As a result, evidence in DS
can be meaningful at a higher level of abstraction without having
to resort to assumptions about the events.

DS fusion was applied to handwriting recognition [67], decision
making [68], face detection [69], landmine detection [60,61,48],
and more [55,70]. One important feature of DS is the ability to cope
with varying levels of precision regarding the information with no
further assumptions needed to represent the information. It also
allows for direct representation of uncertainty of system re-
sponses. However, DS fails to give an acceptable solution to fusion
problems with significant conflict [71,72]. Consequently, many
researchers developed modified Dempster rules to represent the
degree of conflict [70].

DS and Bayesian theories have been studied and compared
extensively [73,57,74]. Both theories have initial requirements.
DS theory requires masses to be assigned to alternatives in a mean-
ingful way, including the unknown state; whereas Bayes theory re-
quires prior probabilities. In general, the results of both methods
may be comparable, but the implementations may require differ-
ent amounts of effort and information. Thus, selecting one ap-
proach over the other usually depends on the extent to which
prior information is available.

Let \( \Theta = \{ \theta_1, \ldots, \theta_k \} \) be a finite set of possible hypotheses, also
referred to as the frame of discernment. The basic belief assign-
mation function, \( m \), a primitive of evidence theory, assigns a value
in \([0,1]\) to every subset \( A \) of \( \Theta \) and satisfies

\[
m(\emptyset) = 0, \quad \text{and} \quad \sum_{A \in \Theta} m(A) = 1. \tag{7}
\]

\( m(A) \) is the belief that supports \( A \), but makes no additional claims
about any of the subsets of \( A \). Two basic belief functions \( m_1 \) and \( m_2 \)
can be combined to obtain the belief mass committed to \( C \subset \Theta \) as
follows [66].

\[
m(C) = m_1(C) \cap m_2(C) = \frac{\sum_{j: j \supseteq C} m_1(A_j) m_2(B_j)}{1 - \sum_{j: j \supseteq C} m_1(A_j) m_2(B_j)}, \quad C \neq \emptyset. \tag{8}
\]

This combination rule is extended to several belief functions by
repeating the rule for new belief functions. The denominator in (8)
is a normalizing factor, which intuitively measures how much \( m_1 \) and
\( m_2 \) are conflicting. This normalization has the effect of com-
pletely ignoring conflict and causing any belief mass associated
with conflict to the null set [71]. Consequently, in the case of a sig-
nificant conflict, this normalization can yield counterintuitive re-
sults. Fortunately, for the application under consideration, alarms
with strongly conflicting evidence are unlikely. This is because all
of the discrimination algorithms considered here use data from the
same sensor (GPR) and try to identify signatures that have a con-
sistent shape.

In some applications, we have prior knowledge about reliability of
the sources. In this case, we can assign their weights before com-
bining their belief functions, resulting in a weighted Demp-
ster–Shafer fusion rule:

\[
m(C) = m_1 \oplus m_2(C) = \frac{\sum_{j: j \supseteq C} w_i m_1(A_j) w_j m_2(B_j)}{1 - \sum_{j: j \supseteq C} w_i m_1(A_j) w_j m_2(B_j)}, \quad C \neq \emptyset. \tag{9}
\]

Since we have classes mine (M) and clutter (C), we build the frame
of discernment as \( \Theta = \{ \emptyset, \{M\}, \{C\}, \{M,C\} \} \). For each individ-
ual algorithm \( i \), we associate a basic belief function \( m^i \) such that

\[
m_i(M) = p^m_i, \quad m_i(C) = p^c_i, \quad \text{and} \quad m_i(M,C) = 1 - p^m_i - p^c_i. \tag{10}
\]

where \( p^m_i \) and \( p^c_i \) are the confidences in the mine and clutter classes
generated by algorithm \( i \). These values are computed from the algo-
rithms' confidence values as follows. First, we separate the training
mine alarms from the clutter alarms and, for each algorithm \( i \), we
compute the cumulative probability distribution of each class, \( C^m_i \)
and \( C^c_i \). Then, we compute \( p^m_i = C^m_i(y_i) \) and \( p^c_i = 1 - C^c_i(y_i) \). Since \( p^m_i \) and \( p^c_i \) are computed independently using the training data of
each class, they are not constrained to sum to 1.

The fusion of the eight algorithms is performed by combining
their basic belief functions using (8) or (9). In the latter case, the
weights are obtained from training data based on individual algo-
richm performance. The final mine confidence is

\[
Conf_{DS} = \sum_{i \in K} m_i(M) - \sum_{i \in K} m_i(C) + K, \tag{11}
\]

where \( K \) is a constant used to ensure that \( Conf_{DS} \geq 0 \) for all test
samples.

4.6. Decision template fusion

Decision template (DT) is a fusion scheme that combines classi-
ifiers outputs by comparing them to a characteristic template for
each class [75]. DT fusion uses all classifier outputs to calculate
the final support for each class, which is in sharp contrast to most
other fusion methods which use only the support for that particu-
lar class to make their decision. The DT approach treats the classi-
ifiers outputs as input to a second-level classifier in some
intermediate feature space, and designs a new classifier for the sec-
ond (combination) level.

DT fusion is computationally simple and does not rely on ques-
tionable assumptions. However, it does not consider the possible
correlation among the individual classifiers. Moreover, its per-
formance may depend on the distribution of the classifiers’ output
which can affect the similarity measure in the intermediate feature
space. DT fusion has been applied to various areas such as time ser-
ies classification [76], biometrics [77], and intrusion detection [78],
and compared with many other fusion techniques. The results are
in general inconclusive, which confirms that there is no fusion
method that outperforms all others in all applications.

Let \( d_i(x) \in [0,1] \) represent the degree of support given by algo-

\[
\text{m}_{j} \] to the hypothesis that \( x \) comes from class \( w_j \) (e.g. the posterior
probability \( P(w_j|x) \)). The outputs of all classifiers are
organized in a decision profile matrix \( \mathbb{D}(x) \). The value in row \( i \) and
column \( j \) of the decision profile matrix is \( d_i(x) \) [75]. Using

Please cite this article in press as: H. Frigui et al., An evaluation of several fusion algorithms for anti-tank landmine detection and discrimination, Inform.
To apply this voting strategy in a supervised learning setting, the Borda based fusion is that it makes no assumptions about the (in a different way) to handwriting recognition [46], and fusion Borda fusion has been applied to landmine detection [80], and algorithm:

\[ \text{Let } \frac{1}{N} \sum_{i \in D_i} \text{DP}(z_i). \]  

(12)

To test a sample \( x \), we construct \( \text{DP}(x) \) and calculate the distance between \( \text{DP}(x) \) and each \( D_i \), using

\[ d_i = \frac{1}{C} \sum_{j=1}^{C} (d_{ij}(x) - d_{ij}(k,j))^2, \]  

(13)

where \( d_{ij}(k,j) \) is the \((k,j)\)th entry in the decision template \( D_i \). The support for class \( w \), offered by combining the \( L \) classifiers, \( \text{ConfBw}(x) \), is then found by measuring the similarity between the current \( \text{DP}(x) \) and \( D_i \) :

\[ \mu_i(x) = 1 - \frac{1}{C} d_i(\text{DP}(x), D_i). \]  

(14)

In the landmine detection application, we use the confidence values of the eight algorithms to construct an \( 8 \times 2 \) decision templates. We let \( d_{1j}(x) = p_m(x) \) and \( d_{2j}(x) = p_c(x) \), where \( p_m \) and \( p_c \) are the mine and clutter probabilities computed as in Section 4.5. The final mine confidence is

\[ \text{ConfBw}(x) = \mu_i(x) \times (1 - \mu_i(x)). \]  

(15)

4.7. Rank-based fusion

This approach is based on the voting method proposed by Borda [79]. Each algorithm ranks all the candidate objects in order of their confidences. In particular, each algorithm \( i \) maps the confidence value of object \( x_j \), \( y[x_j] \), to a rank value \( r \) using

\[ r_i(x_j) = 1 + \sum_{k=1}^N \frac{1}{2} \left( 1 + x_{C} \left( y(x_j), y(x_k) \right) \right). \]  

(16)

In (16), \( x_C \) is the characteristic function that maps a pair in which the first element is greater than or equal to the second to 1 and all other pairs to 0. Similarly, \( x_C \) maps identical pairs to 1. Thus, each object in the training set will have a rank in the interval \[ 1, N \], where \( N \) is the size of the training set.

Let \( x_i \in R, i = 1, \ldots, L \). The weighted Borda fusion of \( L \) algorithms is defined to be weighted sum of the ranks assigned by each algorithm:

\[ \text{ConfBw}(x) = \frac{1}{N \times L} \sum_{i=1}^{L} \sum_{x \in R} \mu_i(x). \]  

(17)

If \( x_i \) is available, (17) is called the Borda count and \( \text{ConfBw}(x) \in [0, 1] \). Borda fusion has been applied to landmine detection [80], and (in a different way) to handwriting recognition [46], and fusion of social choices (voting, evaluation, etc.) The main advantages of the Borda based fusion is that it makes no assumptions about the underlying distributions of the confidence value assignments. In addition, it maps each of the confidence distribution to a uniform distribution, thus providing a reasonable method for combining decision statistics.

To apply this voting strategy in a supervised learning setting, we rank the training set alas shown in (16). Although the algorithm confidences may depend upon the properties of the training set, the ranking process makes no use of such a priori information. Rank values are assigned to test objects using the training set rankings. Thus, if algorithm \( i \) assigns confidence \( x_k \) to object \( k \), we assign rank \( r_i(x_k) \) (the training set rank associated with that algorithm confidence value) to object \( k \).

We have explored weight selection techniques such as Kendall’s rank correlation coefficient [81], coefficient of concordance [82], and weights motivated by gambling theory [83]. All of them outperform unweighted Borda fusion. Exhaustive search can be used to assign weights for small collections of algorithms, but too computationally burdensome for large collections.

Given an assignment of algorithm weights, \( w \), \( \text{ConfBw} \) maps each object to its corresponding confidence. Thus, for each vector \( w \), there is a ROC curve. As in the GEOM detector, we seek to maximize the area under the ROC curve. Consider the function \( AUC(w) \), mapping an algorithm weight assignment to the corresponding area under the ROC curve given by \( \text{ConfBw} \). To identify the best weights to use, we perform gradient ascent on \( AUC(w) \) starting with \( w_i = 1/L \) for all \( i \). The weights are constrained to sum to 1, but they can be either positive or negative.

4.8. Discrete Choquet integral

The Choquet integral has been investigated for information fusion by many researchers [84–89,45,90–93]. This integral defines a family of generally nonlinear aggregation operators on some function of the algorithm confidence values, which we will refer to as a decision statistic. The aggregation operator is defined by the discrete Choquet integral with respect to a non-additive fuzzy measure. As used here, fuzzy measures are real-valued functions defined on sets of algorithms. There are many non-additive measures that can be used with the Choquet integral. The Choquet integral with respect to a specific non-additive measure is a specific aggregation operator such as the mean, median, min, max, trimmed means, Ordered Weighted Averaging operators, and voting operators as well as more complex operators. Many of these operators are already used in fusion. The Choquet integral is a mathematical construct that can be used to optimize the aggregation operator for a specific fusion application.

Discrete fuzzy measures and Choquet integrals are defined as follows [94,86,8,8]:

**Definition 1.** Let \( Y = \{y_1, \ldots, y_L\} \) be any finite set. A discrete fuzzy measure on \( Y \) is a function \( \mu : 2^Y \rightarrow [0, 1] \) with the following properties:

(a) \( \mu(\emptyset) = 0 \) and \( \mu(Y) = 1 \).

(b) Given \( A, B \subseteq Y \), if \( A \subseteq B \) then \( \mu(A) \leq \mu(B) \) (Monotonicity Property).

**Definition 2.** Let \( f : Y \rightarrow [0, 1] \) and let \( \sigma \) denote a permutation such that \( 0 \leq f(y_{\sigma(1)}) \leq \cdots \leq f(y_{\sigma(L)}) \), and let \( A_{\sigma(0)} \) be given by \( A_{\sigma(0)} = \{y_{\sigma(0)}, \ldots, y_{\sigma(L)}\} \). The Choquet integral of \( f \) is:

\[ C_{\mu}(f) = \sum_{i=1}^{L} \mu(A_{\sigma(i)}) f(y_{\sigma(i)}) - f(y_{\sigma(i-1)}) = \sum_{i=1}^{L} f(y_{\sigma(i)}) (\mu(A_{\sigma(i)}) - \mu(A_{\sigma(i-1)})). \]  

(18)

where we take \( f(y_0) = 0 \). \( A_{\sigma(i+1)} \equiv A_{\sigma(i)} \equiv \emptyset \) and \( y_{\sigma(L)} \equiv y_0 \).

In these experiments, algorithm ranks as described in the section on Borda fusion are used as the function \( f \).

Several algorithms have been proposed for learning fuzzy measures [88,95,96]. In this paper, we report the results obtained using a learning algorithm that is based on a Bayesian model that
combines logistic regression with sparsity promoting priors [97]. More specifically, this algorithm seeks to maximize the a-posteriori probability of the measure given the data. The posterior probability of the measure is proportional to the product of the likelihood function and the prior probability of the measure. An exponential prior is assumed on the fuzzy measure parameters. Since the probability of a zero parameter is very high with this prior, it is likely to drive measure parameters to zero in the learning process and potentially eliminate unnecessary algorithms from the fusion. The likelihood function is a binomial distribution, and the MAP estimate is computed using a Gibbs sampling algorithm that is designed to maintain the monotonicity constraints of the fuzzy measure [97].

4.9. Context-dependent fusion

The context-dependent fusion (CDF) approach [51] is motivated by the observation that there is no single algorithm that can consistently outperform all others detectors. For instance, in landmine detection, the relative performance of different detectors can vary significantly depending on the mine type, geographical site, soil and weather conditions, and burial depth.

The training part of CDF has two main components: Context Extraction, and Algorithm Fusion. In Context Extraction, the features extracted by the different algorithms are combined, and a clustering algorithm is used to partition the training signatures into groups of similar signatures, or contexts, and learn the relevant features within each context. It is assumed that signatures that have similar response to different algorithms share some common features, and would be assigned to the same cluster. The Algorithm Fusion component assigns an aggregation weight to each detector in each context based on its relative performance within the context. To test a new signature using CDF, each detector extracts its set of features and assigns a confidence value. Then, the features are used to identify the best context, and the aggregation weights of this context are used to fuse the individual confidence values.

We should note here that CDF is an alternative approach to data fusion that is local, and that adapts the fusion method to different regions of the feature space. It has been applied to landmine detection [51] using simple linear aggregation. However, any of the fusion methods outlined earlier could be integrated into this approach.

Table 1

<table>
<thead>
<tr>
<th>Fusion Alg.</th>
<th>Assumption</th>
<th>Input</th>
<th>Local/ global</th>
<th>Considers classifiers correlation</th>
<th>Considers subsets of classifiers</th>
<th>Aggregation weights</th>
<th>Requires training</th>
<th>Optimized criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes</td>
<td>Mixture of Gaussian</td>
<td>Conf.</td>
<td>Global</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
<td>Log likelihood</td>
</tr>
<tr>
<td>Mahalanobis distance</td>
<td>Gaussian distribution</td>
<td>Conf.</td>
<td>Global</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
<td>Average FAR for PD ∈ [92%, 96%]</td>
</tr>
<tr>
<td>Dempster–Shafer</td>
<td>N/A</td>
<td>Conf.</td>
<td>Global</td>
<td>No</td>
<td>Positive</td>
<td>No</td>
<td>Yes</td>
<td>Distance to decision template</td>
</tr>
<tr>
<td>Decision template</td>
<td>N/A</td>
<td>Conf.</td>
<td>Global</td>
<td>No</td>
<td>Positive/ negative</td>
<td>No</td>
<td>Yes</td>
<td>Area under ROC</td>
</tr>
<tr>
<td>Borda</td>
<td>N/A</td>
<td>Rank</td>
<td>Global</td>
<td>No</td>
<td>Positive</td>
<td>No</td>
<td>Yes</td>
<td>Posterior prob. of the measures</td>
</tr>
<tr>
<td>Fuzzy integral</td>
<td>N/A</td>
<td>Rank</td>
<td>Global</td>
<td>Yes</td>
<td>Positive</td>
<td>No</td>
<td>Yes</td>
<td>Class dist. overlap</td>
</tr>
<tr>
<td>Context-dependent</td>
<td>N/A</td>
<td>Conf. + Feat</td>
<td>Local</td>
<td>No</td>
<td>Positive</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

The above fusion methods were selected for evaluation and comparison for the landmine detection application because they have very distinctive properties. For instance, one method (CDF) is local and adapts the detectors’ worthiness to different data subspaces. The other methods are global and assign a degree of worthiness to each detector that is averaged over the entire training data. Also, some fusion methods operate on the detectors’ confidence values of the alarms, others (Borda and fuzzy integral) operate on the ranks of the alarms, and others (CDF) require both confidence values and features used by the classifiers. Another main difference between these fusion methods is the way they are trained. Some methods use straightforward training (e.g. Decision template and Bayes), while others (e.g. fuzzy integral) use more elaborate training algorithms. Moreover, the trainable methods use different optimization criteria. For instance, some try to maximize the area under the ROC, while others minimize the overlap between the distribution of the confidence values in the classes of mines and clutter. These algorithms were developed by various subsets of the authors. Maximal performance for each fusion algorithm was always the goal of the algorithm developer. As is always the case, it is possible that better performance could be found with any of the tested approaches, such as Dempster–Shafer, for example. The characteristics of the different fusion methods are summarized in Table 1.
into three categories: anti-tank metal (ATM), anti-tank with low metal content (ATLM), and simulated mines (SIM). The targets were buried up to 6 in. deep. Multiple data collections were performed at each site at different dates, covering a ground area of \(14,813 \text{ m}^2\), resulting in a large and diverse collection of mine and false alarm signatures. False alarms arise as a result of radar signals that present a mine-like character. Such signals are generally said to be a result of clutter. In this experiment, clutter arises from two different processes. One type of clutter is emplaced and surveyed in an effort to test the robustness of the algorithms. Other clutter result from human activity unrelated to the data collection or as a result of natural processes. We refer to this second kind of clutter as non-emplaced. Non-emplaced clutter includes objects discarded or lost by humans, soil inconsistencies and voids, stones, roots and other vegetation, as well as remnants of animal activity.

The statistics of the data are shown in Table 2. The data collected from Sites B and D have emplaced buried clutter. Although the lanes at Sites A and C are prepared, they still contain non-emplaced clutter objects. Both metal and non-metal non-emplaced clutter objects such as ploughshares, shell casings, and large rocks have been excavated from these sites. The emplaced clutter objects include steel scraps, bolts, soft-drink cans, concrete blocks, plastic bottles, wood blocks, and rocks. In all, there are 12 collections having 19 distinct mine types. Many of these mine types are present at several sites. The prescreener detected 1593 mines encountered in the data, yielding a 97.9% probability of detection. It rejected 161 of 211 emplaced clutter objects encountered, and yielded a total of 2435 false alarms associated with non-emplaced clutter objects. The number, type, and burial depth of the mines are given in Table 3. As it can be seen, the mines buried at 1 in. through 6 inches occupy 87.5% of the total targets encountered vs. 12.5% surface-laid or flush-buried mines.

Table 2
Statistics of the dataset.

<table>
<thead>
<tr>
<th>Site</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. collections</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>No. mine types</td>
<td>9</td>
<td>15</td>
<td>9</td>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>No. mine alarms</td>
<td>183</td>
<td>821</td>
<td>62</td>
<td>494</td>
<td>1560</td>
</tr>
<tr>
<td>No. clutter encounters</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>196</td>
<td>211</td>
</tr>
<tr>
<td>No. clutter alarms post prescreener</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>46</td>
<td>50</td>
</tr>
<tr>
<td>Area (m²)</td>
<td>14,813</td>
<td>15,631</td>
<td>4054</td>
<td>7310</td>
<td>41,808</td>
</tr>
</tbody>
</table>

Table 3
Number of metal and plastic cased mines and mine simulants and their burial depths.

<table>
<thead>
<tr>
<th>Depth</th>
<th>–1 in.</th>
<th>0 in.</th>
<th>1 in.</th>
<th>2 in.</th>
<th>3 in.</th>
<th>4 in.</th>
<th>5 in.</th>
<th>6 in.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metal</td>
<td>12</td>
<td>37</td>
<td>124</td>
<td>68</td>
<td>151</td>
<td>34</td>
<td>119</td>
<td>77</td>
<td>777</td>
</tr>
<tr>
<td>Low-metal</td>
<td>6</td>
<td>92</td>
<td>90</td>
<td>204</td>
<td>122</td>
<td>134</td>
<td>47</td>
<td>76</td>
<td>616</td>
</tr>
<tr>
<td>Simulants</td>
<td>48</td>
<td>0</td>
<td>20</td>
<td>47</td>
<td>23</td>
<td>29</td>
<td>0</td>
<td>0</td>
<td>167</td>
</tr>
<tr>
<td>Total</td>
<td>66</td>
<td>129</td>
<td>234</td>
<td>319</td>
<td>296</td>
<td>197</td>
<td>166</td>
<td>153</td>
<td>1560</td>
</tr>
</tbody>
</table>

5.4. Results and analysis

5.4.1. Individual detection algorithms

First, we compare the performance of the individual detectors and justify the need to fuse their results to improve the overall performance of the system. Fig. 3 displays the ROC's obtained by applying the seven detection algorithms and the prescreener to the entire data collection. As it can be seen, the EHD detector has the best overall performance. However, this does not necessarily mean that the EHD is consistently the best algorithm. For instance, Fig. 4a displays the results averaged over site A of the collection only. For this subset, the EHD is the best algorithm and the HMM
is the second best one. However, in Fig. 4b, which displays the results averaged over site B only, the HMM is the best algorithm and EHD is the second best one.

Thus, there is no single algorithm that can consistently outperform all others detectors. In fact, the relative performance of different detectors can vary depending on the geographical site and soil and weather conditions. Moreover, even within the same site, the relative performance of the different algorithms can vary significantly depending on the mine type, burial depth, and other unknown factors. To illustrate this, we compare the output of the HMM and EHD detectors for a small subset of alarms extracted from the same site in Fig. 5. For instance, the highlighted region (R1) in Fig. 5a includes mainly clutter signatures where the HMM algorithm outperforms the EHD (lower HMM confidence values). On the other hand, for the same subset, region (R2) includes mainly mine signatures where the EHD detector outperforms the HMM (higher EHD confidence). Fig. 5b highlights two other regions for another geographical site.

5.4.2. Fusion results

Our objective is to evaluate a set of fusion methods to combine the output of several landmine discrimination algorithms to determine their suitability for use in an automated detection system in a variety of locations and under different environments. In addition to the performance of these fusion methods, we are also interested in their scalability with respect to the number of discrimination algorithms. Thus, we compare these methods when 4, 6, and 8 discrimination algorithms are considered.

Fig. 6 displays the results of the seven fusion algorithms when only four discrimination algorithms (EHD, HMM, Spect, and NUKE) are combined.

Fig. 4. Performance of the eight detectors on: (a) Site A only; and (b) Site B only.

Fig. 5. Comparison of the EHD and HMM outputs for several mine (green dots) and clutter (red stars) signatures extracted from: (a) a subset of Site A; and (b) a subset of Site B. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Comparison of seven fusion methods when four discrimination algorithms (EHD, HMM, Spect, and NUKE) are combined.
are fused. We also include the ROC of the EHD (best overall discrimination algorithm) as a reference. As it can be seen, the ROC's of all fusion methods are clustered together, and thus all methods have comparable performances. All fusion methods improve the PD results over the best discrimination algorithm by an average of 10% for FAR around 0.0007. At low PD (<80%), the Mahalanobis distance based fusion results are not as good as the other methods.

This is due mainly to the fact that one single Gaussian component may not be sufficient to model the distribution of the confidence values of the individual discriminators in the four-dimensional confidence space. The Bayes-based method, which is similar to the distance based, does not exhibit this behavior because multiple Gaussian components ($M$ estimated to be 4) were used to model the distribution of each class. It is also interesting to note that the distance based fusion outperforms Bayes at higher PD. This is because the former method is optimized to minimize the average FAR for PD $\in [92\%, 96\%]$.

Fig. 7 displays the results of the seven fusion algorithms when only six discrimination algorithms (EHD, HMM, Spect, NUKE, Geom, and TFCM) are fused. First, we notice the addition of two discrimination algorithms did not improve the results of any of the fusion methods. Two possible reasons may explain this behavior. First, the added discrimination algorithms (TFCM and Geom) are based on edge, texture, and geometric features that are already used (in a different way) by the other discrimination algorithms. Second, it is possible that for the data collection that was used, it is not possible to improve the results further.

Comparing the results in Fig. 7 to those in Fig. 6, we observe that for some fusion methods, the performance has degraded. In particular, the performance of the Dempster–Shafer (DS) and the decision template (DT) methods have dropped significantly at low PD (<80%) and have become even worse than the EHD discriminator.

Investigation of this problem has revealed that these two fusion methods generate confidence values that have a distribution close to binary. This behavior is due to the way the basic belief functions are aggregated (refer to Eq. (9)). In particular, adding more algorithms will require more multiplications. For the DT method, the dimension of the decision template matrix increases, and this may drive the distributions in (13) to a bimodal distribution. Due to these nearly binary distributions, weak mines will be assigned confidence values close to zero, and this would explain the lower PD at low FAR. Also, strong false alarms will be assigned confidence values close to 1, and this would explain the relatively lower PD at higher FAR.

Fig. 7. Comparison of seven fusion methods when six discrimination algorithms (EHD, HMM, Spect, NUKE, Geom, and TFCM) are combined.

Fig. 8 compares the results of the seven fusion algorithms when eight discrimination algorithms (EHD, HMM, Spect, NUKE, Geom, TFCM, GFIT, and GMRF) are fused. First, we note that the performance of the DT and DS degraded further as the confidence values become closer to binary. Second, the performance of all other fusion methods (except CDF) have degraded compared to the fusion of four algorithms only. This may be due to the fact that the four added algorithms have lower performances (refer to Fig. 3), and when all eight algorithms are fused globally, the added algorithms have a negative impact. Third, we note that the dependency assumption does not seem to be an issue. In fact, the two best fusion methods (CDF and Borda) assume that the eight discrimination algorithms are independent.

The Borda count fusion is the second best method, and does not seem to be affected by the addition of discrimination algorithms. This is due to the fact that this method allows for negative aggregation weights as long as they improve the area under the ROC. Thus, as we add more discrimination algorithms (with worse overall performance), this method will assign negative (or zero) weights to these algorithms.

The CDF has the best overall performance. Moreover, the addition of discrimination algorithms did not degrade its performance. In fact, for certain FAR values, its performance has improved. This is due to the fact that this method is local and strives to take advantage of the different detectors in different contexts. For any cluster (or context) the detectors are ranked based on the overlap between the mine and clutter confidence distribution. This ranking can ignore (by assigning low aggregation weights) many of the discrimination algorithms. It could also assign a significant weight to discrimination algorithms that are good for the given context, but globally, are not as good as other algorithms. We have observed that on average, this fusion assigns significant aggregation weights to 3–5 discrimination algorithms. These algorithms differ from one cluster to another.

Finally, we should note the fuzzy integral approach is trained using a learning algorithm that combines logistic regression with sparsity promoting priors. Thus, it is designed to ignore individual discrimination algorithms that do not improve the results. However, the results do not seem to support this. This may be due to the fact that the number of parameters increases exponentially as we increase the number of algorithms. Thus, the search for the optimal parameters becomes more complex and may lead to suboptimal solutions.
6. Conclusions

We have presented results of an evaluation of several fusion methods to combine the output of several anti-tank landmine discrimination algorithms. Our objective was to determine the suitability of these methods for use in an automated detection system in a variety of locations and under different environments.

Our extensive research and testing in this application has revealed that algorithm performances for buried anti-tank landmine detection are strongly dependent upon a variety of factors that are not well understood. It is typically the case that one algorithm may perform well in one setting and not so well in another. Thus, in order to achieve a reliable and robust detection system, several distinct detection algorithms need to be developed and fused. Therefore, in addition to the performance of the different fusion methods, we are also interested in their scalability with respect to the number of discrimination algorithms. In particular, their ability to take advantage of discrimination algorithms that perform well for only a small subset of the data without being affected by their weakness. To investigate this, we have compared the seven fusion methods when 4, 6, and 8 discrimination algorithms are considered.

Our experimental results show that although the fusion algorithms were all quite similar when a small number of algorithms were fused, the performance was more varied as the number of algorithms increased. Context-dependent fusion appears to be an excellent approach that should be investigated in more detail in future work. Aggregation operators that are allowed to use negative weights appear to perform better than those that do not. Sparsity promoting priors do not necessarily lead to better performance as the number of algorithms increases. The tradeoff between promoting sparsity and computational complexity is difficult to control. Fusion algorithms that tend to binarize confidence values as the number of inputs increases also degrade as a function of the number of algorithms fused. The assumption that the individual detectors are statistically independent does not seem to be a significant factor in affecting the performance of the fusion methods. However, this may be an important issue should the need to reduce the overall computational requirements of the system arise.

Future work will look at integrating the Bayes, Dempster-Shafer, and the Choquet fusion methods within the context-based fusion concept.

Acknowledgment

The authors thank R. Harmon, R. Weaver, P. Howard, and T. Donzelli for their support of this work, E. Rosen and L. Ayers of IDA for useful software and insight, We also thank L. Carin, L. Collins and P. Torrione of Duke University and NIITEK, Inc., for their insights, cooperation, discrimination algorithms, and data. This work was supported in part by NSF Awards No. CBET-0730802 and CBET-0730848, ONR Award Number N00014-05-01788, ARO and ARL Cooperative Agreement Number DAAD19-02-2-0012 and Grant Number DAAB15-02-D-0003. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, Office of Naval Research, Army Research Laboratory, or the US Government.

References


P. Torrone, personal communication.


M.G. Kendall, A new measure of rank correlation, Biometrika 30 (1/2) (1938) 81–93.


