Final Report
SBIR A99-036 *Surface and Buried Mine Detection with Variance-Based Multispectral Data Fusion*
(Phase I contract period: Dec. 2000 – Jun. 2000)

10 November 2000

Gary Key (e-mail: garyk@fti-net.com)
Mark Schmalz (e-mail: mssz@cise.ufl.edu)

Prepared for
US ARMY/CECOM RD-NV-SD-CM(MAK) (George Maksymonko)
Under Contract # DAAB07-00-C-H001

REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
15-08-2000	Final Report	18-12-1999 to xx-xx-2000

4. TITLE AND SUBTITLE Surface and Buried Mine Detection with Variance-Based Multispectral Data Fusion SBIR-A99-036 Unclassified	**5a. CONTRACT NUMBER** DAAB07-00-C-H001
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Key, Gary ; Schmalz, Mark ;	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME AND ADDRESS Frontier Technology, Inc. 6785 Hollister Avenue Goleta , CA 93117	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS US Army CECOM Attn: RD-NV-SC-CM (MAK) 10221 Burbeck Road ,	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
A
PUBLIC RELEASE

US Army CECOM
Attn: RD-NV-SC-CM (MAK)
10221 Burbeck Road
,

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The detection of surface or buried landmines is complicated by effects of occlusion due to overlying vegetation and confusion of the mine spectral response due to overlying obscurants. Such effects often cause mine detection systems that are strictly model-based to fail in field practice due to brittleness resulting from lack of input coverage. University of Florida has assisted Frontier Technology, Inc. in analyzing, developing, and implementing prototype algorithms and software for buried land mine detection using infrared imagery. This technique detects statistical differences between target and background regions and produces and estimate of target probability at a given location and, where possible, an estimate of target identity. University of Florida has assisted FTI in analyzing its Tabular Nearest Neighbor Encoding paradigm that has been highly successful in detecting small targets imagery and other signatures. TNE has been applied to mine detection problems to investigate its utility in producing increased probability of detection and decreased rate of false alarms. This technique has been found to be useful in a wide variety of military and commercial applications.

15. SUBJECT TERMS

mine detection; automatic pattern recognition; minefield detection

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Kye, Shirley skye@dtic.mil
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Public Release	30	**19b. TELEPHONE NUMBER** **International Area Code** **Area Code Telephone Number** 703 767-9111 **DSN** 427-9111

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
15 August 2000	Final Report	18 Dec 1999 – 18 June, 2000

4. TITLE AND SUBTITLE

Final Report
SBIR A99-036 *Surface and Buried Mine Detection with Variance-Based Multispectral Data Fusion*

5a. CONTRACT NUMBER DAAB07-00-C-H001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gary Key, Mark Schmalz

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

FRONTIER TECHNOLOGY, INC.
6785 Hollister Avenue
Goleta, CA 93117

8. PERFORMING ORGANIZATION REPORT NUMBER

00-

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

US Army CECOM
Attn: RD-NV-SD-CM (MAK)
(George Maksymonko)
10221 Burbeck Road

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The detection of surface or buried landmines is complicated by effects of occlusion due to overlying vegetation and confusion of the mine spectral response due to overlying obscurants. Such effects often cause mine detection systems that are strictly model-based to fail in field practice due to brittleness resulting from lack of input coverage.

University of Florida has assisted Frontier Technology, Inc. in analyzing, developing, and implementing prototype algorithms and software for buried land mine detection using infrared imagery. This technique detects statistical differences between target and background regions and produces an estimate of target probability at a given location and, where possible, an estimate of target identity.

University of Florida has assisted FTI in analyzing its Tabular Nearest Neighbor Encoding paradigm that has been highly successful in detecting small targets imagery and other signatures. TNE has been applied to mine detection problems to investigate its utility in producing increased probability of detection and decreased rate of false alarms. This technique has been found to be useful in a wide variety of military and commercial applications.

15. SUBJECT TERMS

Mine detection
Minefield detection
automatic pattern recognition

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gary Key
unclassified	unclassified	unclassified	Public Release	30	19b. TELEPHONE NUMBER (Include area code) (321) 277-8396

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Table of Contents

Abstract

The detection of surface or buried landmines is complicated by effects of occlusion due to overlying vegetation and confusion of the mine spectral response due to overlying obscurants. Such effects often cause mine detection systems that are strictly model-based to fail in field practice due to brittleness resulting from lack of input coverage.

Frontier Technology and the University of Florida have made progress in analyzing, developing, and implementing prototype algorithms and software for buried land mine detection using infrared imagery. The techniques involved detect statistical differences between target and background regions and produce an estimate of target probability at a given location and, where possible, an estimate of target identity.

University of Florida has assisted FTI in analyzing its Tabular Nearest Neighbor Encoding paradigm that has been highly successful in detecting small targets imagery and other signatures. TNE has been applied to mine detection problems to investigate its utility in producing increased probability of detection and decreased rate of false alarms. This technique has been found to be useful in a wide variety of military and commercial applications.

1. Introduction

Frontier Technology (FTI) and the University of Florida (UF) are pleased to submit this final report pursuant to Contract #DAAB07-00-C-H001 resulting from award to FTI on Phase I of ARMY SBIR Solicitation 99.2, Item A99-036 entitled "Landmine Detection". The successful application of machine vision and pattern recognition techniques to automated detection and recognition of surface or buried mine continues to be an elusive goal. Machine vision systems currently operate under tightly specified constraints on lighting, object positioning or manipulation, and object pose that are not practical in realistic battlefield imaging scenarios. Past attempts at adapting machine vision algorithms for automated target recognition (ATR) have failed due to lack of fundamental knowledge of the human visual system. However, research has revealed that ATR processing based on object outlines, boundaries, and contours, as well as color and local variance information, when accompanied by higher-level recognition of feature groups (e.g., co-occurrence of color, texture, size, shape, and statistical properties) can enhance the probability of detection (Pd) in ATR of mine targets while simultaneously decreasing the rate of false alarms (Rfa).

In response to this situation, FTI and UF have leveraged UF's previous research and development experience in (1) detection and recognition of mines and mine-like targets, as well as (2) detection and characterization of patterned and non-patterned minefields, to develop algorithms for recognition of a wide variety of surface or buried mines from standoff or airborne imagery. In Phase I of the proposed effort, the FTI team has (a) assembled a sample database of surface mine and minefield imagery, (b) applied algorithms, software, and imagery for mine detection and recognition, (c) investigated the effect of noise and sensor error on surface and buried mine detection and recognition, and (d) enhanced UF's and FTI's existing algorithms for minefield detection and characterization. UF and FTI originally proposed to incorporate FTI's TNE pattern recognition algorithm and (if feasibility analyses so indicated) UF's morphological neural network (MNN) based pattern recognition paradigm to detect, classify, and provide preliminary recognition estimates of a variety of mines and mine-like targets in various clutter situations (e.g., exposed, partially covered, or buried). However, at the direction of the Army POC Mr. George Maksymonko the team concentrated its effort on (1) analysis of the double

gated filter, (2) development of algorithms for finding mines in patterned minefields after some (but not all) of the mines have been detected, and (3) a more exotic approach that involves a statistical form of vector quantization.

The result of this work is prototype software (in C++, FORTRAN and Matlab m-codes) that detects mines and minelike targets on a background that is statistically different from the targets, and uses this information to detect potential mines or minelike targets in surface or buried minefields.

The technique has been developed using near-infrared (NIR) imagery provided by the Army POC. Visible-wavelength imagery obtained by UF and Naval Surface Warfare Center (NSWC, Dahlgren Division), using a wide variety of landmines with a calibrated multispectral camera and reflectance targets, was employed for reference purposes.

2. Adaptive Double-Gated Filter

In previous research, we developed a strategy for feature-based detection of surface and buried mines and minefields that implemented an instance of the detect-before classify paradigm for automated target recognition (ATR) [2,3]. Our method is based on five proven innovations: (a) a well-established suite of ATR filters specifically developed for mine detection [4-6]; (b) automated performance analysis of mine detection and ATR algorithms using well-accepted metrics such as Pd and Rfa [7-10]; (c) semi-automatic minefield detection and characterization algorithms [11,12]; and (d) a data fusion paradigm such as morphological neural networks, which significantly improve the performance, accuracy, and information capacity of trainable classifiers [13,14].

Under prior DoD-sponsored research, the FTI team has successfully developed and tested a suite of four ATR filter classes specifically designed for surface and underwater mine detection in imagery estimate altitude, and ground speed (as available). Alternatively, target shape and size can at times be estimated indirectly from world knowledge of ground features such as size of known groundcover, scale of surf breaks in near-shore applications, or terrain features such as vehicle tracks or bridges.

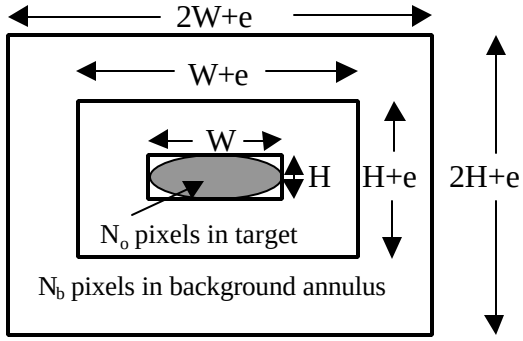


Figure 1. Box notation for adaptive double gated filter. All boxes are on a common center.

UF's filters do not require precise knowledge of target size or shape, and can be adapted to time-varying constraints on target geometric and statistical parameters. A further advantage of UF's approach is that the geometric/statistical filters compute their own statistical parameters adaptively, and thus do not suffer from lack of robustness in model-based ATR.

2.1. Mathematical Formulation

In particular, UF's adaptive double-gated filter (ADGF) assumes that a truthed target in a

monochromatic image $\mathbf{a} \in \mathbf{F}^{\mathbf{X}}$ is surrounded by three boxes, as shown in Figure 1, where the truthed target, depicted ellipse, has N_o pixels and is surrounded by a bounding box of size HxW pixels. Additionally, N_b background pixels are contained in the (2H+e)x(2W+e)-pixel box but not within the (H+e)x(W+e)-pixel box. Assume that the background greylevel ranges from 1 to r_b and the truthed target has an intensity range from one to r_o. Note that the boxes need not be rectangular, but can be any shape, provided that they are annularly nested on a common center. The adaptive nature of the ADGF arises from the fact that the boxes can vary in size and shape with image statistics, sensor look angle, and sensor altitude, as discussed in Section 2.3.

Given a target point **x** within the truthed target, denote the NxW-pixel box that surround the target as U(y), which is a subset of X. Similarly, denote the (H+e)x(W+e)-pixel box as **V**(**y**), and the (2H+e)x(2W+e)-pixel box as **W**(**y**), which are both subsets of **X**. The target impulse response at **y** is given by

$$\mathrm{DGF}_{\mathrm{out}}(\mathbf{y}) = \mathrm{TIR}^2(\mathbf{y}) = \frac{\left(\frac{1}{N_o} \sum_{\mathbf{u} \in \mathbf{U}(\mathbf{y})} \mathbf{a}(\mathbf{u}) - \frac{1}{N_b} \sum_{\mathbf{x} \in \mathbf{W}(\mathbf{y}) \setminus \mathbf{V}(\mathbf{y})} \mathbf{a}(\mathbf{x}) \right)^2}{\frac{1}{N_b} \sum_{\mathbf{x} \in \mathbf{W}(\mathbf{y}) \setminus \mathbf{V}(\mathbf{y})} \mathbf{a}(\mathbf{x})^2 - \left(\frac{1}{N_b} \sum_{\mathbf{x} \in \mathbf{W}(\mathbf{y}) \setminus \mathbf{V}(\mathbf{y})} \mathbf{a}(\mathbf{x}) \right)^2} = \frac{(\bar{r}_o - \bar{r}_b)^2}{\sigma_b^2} ,$$

where $\bar{r}_b$ and $\bar{r}_o$ denote mean values, with the standard deviation of the background denoted by s_b. The optimal ADGF output is achieved when TIR^2 is maximized.

2.2. Image Algebra Formulation

From the preceding equation, it is easily verified that, given the linear convolution operation $\oplus : \mathbf{R}^{\mathbf{X}} \times \left(\mathbf{R}^{\mathbf{X}}\right)^{\mathbf{X}} \to \mathbf{R}^{\mathbf{X}}$, and templates $\mathbf{s}, \mathbf{t} \in \left(\mathbf{R}^{\mathbf{X}}\right)^{\mathbf{X}}$ defined by their weights as follows:

$$\mathbf{s}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \mathbf{W} \setminus \mathbf{V} \\ 0 \text{ otherwise} \end{cases} \quad \text{and} \quad \mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \mathbf{U} \\ 0 \text{ otherwise} \end{cases}, \ \mathbf{y} \in \mathbf{X},$$

such that the resultant TIR^2 map, denoted by **c**, is expressed in image algebra [15] as

$$\mathbf{c} = \frac{\left[(\mathbf{a} \oplus \mathbf{t}) - (\mathbf{a} \oplus \mathbf{s})\right]^2}{\left(\mathbf{a}^2 \oplus \mathbf{s}\right) - (\mathbf{a} \oplus \mathbf{s})^2} = \frac{\left[\mathbf{a} \oplus (\mathbf{t} - \mathbf{s})\right]^2}{\left(\mathbf{a}^2 \oplus \mathbf{s}\right) - (\mathbf{a} \oplus \mathbf{s})^2} .$$

2.3. Implementational Issues

Computation of the ADGF response is simplified by examining the right member of the TIR^2 expression in Section 2.1, which states:

$$\mathrm{TIR}^2 = \left(\frac{\bar{r}_o - \bar{r}_b}{\sigma_b} \right)^2 .$$

Thus, a simple implementation of the ADGF would (1) compute the target and background means $\bar{r}_b$ and $\bar{r}_o$, (2) compute the background standard deviation σ_b, then (3) compute the squared quotient.

Determination of the target mean is straightforward, but computation of the background mean and standard deviation requires partitioning of the background annulus shown in Figure 1. First, we subdivide the annulus into top, bottom, left, and right parts, then insert the pixel values located in these parts into a list *L*. By taking the mean and standard deviation of the values in *L*, we then obtain $\bar{r}_b$, $\bar{r}_o$, and σ_b, from which TIR^2 can be calculated. This process is typically repeated for each point **x** in the source image domain **X**, but can be made more efficient, as follows.

Since the majority of the pixel values in $L(\mathbf{x}_i)$ are found in the application of the ADGF to the

adjacent domain point, i.e., in $L(\mathbf{x}_{i+1})$, it is possible to build the four partitions of the background annulus using a queue for each row of each partition. This technique is also used when implementing the target box. From Figure 1, it is readily verified that, for each horizontal shift of the ADGF domain, 4H + 2e pixels are enqueued and dequeued, yielding a total of 8H + 4e I/O *enqueue* and *dequeue* operations. This is significantly fewer than the $4HW + 2e(W+H) + e^2$ I/O operations required if each pixel value in the target box and background annulus is replaced at each new position of the double-gated filter.

A further implementational consideration involves adaptive parameterization of the ADGF to reflect a sensor's instantaneous field-of-view (IFOV). Given sensor look angle zenith ϕ and azimuth θ, sensor altitude a, and a right-hand Cartesian coordinate system, a pixel subtense of s_c in the focal plane corresponds to a radial displacement h from nadir in the target plane is related to the camera focal length f_c as

$$\phi = \tan^{-1}(s_c / f) .$$

Assuming unitary camera lens magnification, by similar triangles we have

$$h = a\, s_c / f_c .$$

Thus, if sensor altitude a increases by a factor $f_a > 0$, all other parameters being held constant, then $\tan(\phi)$ varies as $1/f_a$. Since θ and h are assumed to be constant, the number of pixels subtended at the focal plane varies as $1/f_a$, since $s_c = f_c \tan(\phi)$. Thus, the focal plane magnification factors (or minification, if $0 < f_a < 1$) in the x- and y-axis directions are given by

$$f_x = \cos(\theta) / f_a \qquad \text{and} \qquad f_y = \sin(\theta) / f_a .$$

This means that the filter box dimensions W (x-direction), H (y-direction), and e can be revised according to f_x and f_y to cause the ADGF to adapt to changes in sensor altitude.

From the preceding development, if ϕ is multiplied by factor f_ϕ, then the W and H settings vary by factor f_c, which is determined as follows. From trigonometry,

$$\tan(kx) = \frac{\tan((k-1)x) - \tan(x)}{1 - \tan((k-1)x)\tan(x)} .$$

Since h varies as $\tan(\phi)$, all parameters except ϕ being held constant, the change in h, denoted by f_h, will equal f_c. Thus, $f_c = \tan(f_\phi \phi) / \tan(\phi)$, which can be expressed in terms of the preceding equation for $\tan(kx)$ as

$$f_c = \frac{(\sin((f_\phi - 1)\phi)\cos(\phi) + \sin(\phi)\cos((f_\phi - 1)\phi))\cos(\phi)}{\cos(\phi)\sin(\phi)\cos((f_\phi - 1)\phi) - \sin((f_\phi - 1)\phi) + \sin((f_\phi - 1)\phi)\cos^2(\phi)} ,$$

which is resolved to the x- and y-axes as

$$f_x = f_c \cos(\theta) \qquad \text{and} \qquad f_y = f_c \sin(\theta) .$$

As a result, one can compute revised values of W and H given variations in sensor altitude a and look angle ϕ, as

$$W' = W f_x \qquad \text{and} \qquad H' = H f_y .$$

The offset parameter e can be resolved into components e_x and e_y that can be similarly compensated for sensor look angle and elevation.

3. Analysis of the Adaptive Double-Gated Filter

In this section, we analyze the complexity and worst-case computational error incurred by the ADGF. The analysis expands upon research presented in Reference 5, and has been developed for the noisy imagery utilized in this study.

3.1. Complexity Analysis

From Section 2 and Figure 1, recall that the ADGF is applied at a target point **y** within the HxW-pixel box that surrounds the target, whose domain is denoted as **U**(y), a subset of the source image domain **X**. Similarly, the (H+e)x(W+e)-pixel box that delimits the inner extent of the background annulus is denoted as **V**(**y**), and the (2H+e)x(2W+e)-pixel box that delimits the outer

extent of the annulus is denoted by **W**(**y**), which are both subsets of **X**. The target impulse response at **y** is given by

$$DGF_{out}(\mathbf{y}) = TIR^2(\mathbf{y}) = \frac{\left(\frac{1}{N_o}\sum_{\mathbf{u}\in\mathbf{U}(\mathbf{y})}\mathbf{a}(\mathbf{u}) - \frac{1}{N_b}\sum_{\mathbf{x}\in\mathbf{W}(\mathbf{y})\setminus\mathbf{V}(\mathbf{y})}\mathbf{a}(\mathbf{x})\right)^2}{\frac{1}{N_b}\sum_{\mathbf{x}\in\mathbf{W}(\mathbf{y})\setminus\mathbf{V}(\mathbf{y})}\mathbf{a}(\mathbf{x})^2 - \left(\frac{1}{N_b}\sum_{\mathbf{x}\in\mathbf{W}(\mathbf{y})\setminus\mathbf{V}(\mathbf{y})}\mathbf{a}(\mathbf{x})\right)^2} = \frac{(\bar{r}_o - \bar{r}_b)^2}{\sigma_b^2},$$

where $\bar{r}_b$ and $\bar{r}_o$ denote background and target means and s_b denotes background standard deviation.

Computation of the target mean requires $|\mathbf{U}(\mathbf{y})| - 1 = HW - 1$ additions and one division operation, which can be implemented as multiplication by the fraction 1/HW. Letting $N_b = |\mathbf{W}(\mathbf{y}) \setminus \mathbf{V}(\mathbf{y})|$, computation of the background mean requires $N_b - 1$ additions and one multiplication, similar to the target mean. In the case of a boxy ADGF domain as described above, we have that

$$\begin{aligned} N_b &= (H+e)(W+e) - (2H+e)(2W+e) \\ &= 4HW + 2(We + He) + e^2 - (HW + We + He + e^2) \\ &= 3HW + e(H+w) . \end{aligned}$$

Computation of the background standard deviation s_b requires computation of the background mean, N_b subtractions (additions) and multiplications to compute the squared differences, $N_b - 1$ additions to compute the sum of squares, with one multiplication and one square root. A subtraction, one division (multiplication), and one multiplication are required to compute TIR^2. Thus, at each point **y** in **X**, if values in the target box and background annulus are not buffered as described in Section 2, then the computational budget shown in Table 1 applies.

Table 1. Computational budget for ADGF without buffering of pixel values.

Stage of Computation	**Additions**	**Multiplications**	**Roots**
Target Mean r_o	HW – 1	1	0
Background Mean r_b	3HW+e(H+W)-1	1	0
Background Stdev σ_b	6HW+2e(H+W)-1	3HW+e(H+W)+1	1
$TIR^2(\mathbf{y})$	1	2	0
Total Work Required	10HW+3e(H+W)-2	3HW+e(H+W)+5	1

As stated in the cursory analysis of Section 2, it is easily verified that the ADGF work requirement W is **O**(HW) in multiplications and additions. As H and W are varied to compensate for sensor altitude and look angle, W will vary accordingly, increasing supralinearly as look angle increases linearly.

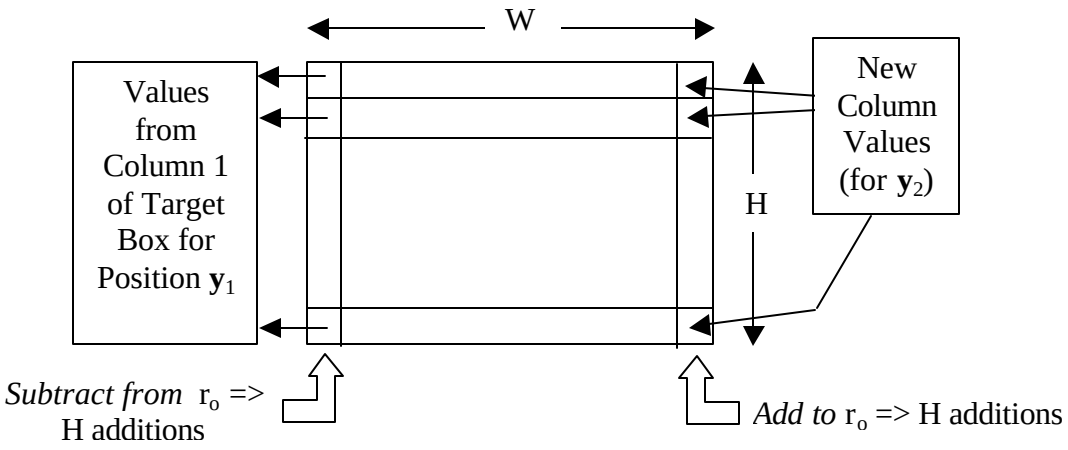


Figure 2. Buffering scheme for efficient computation of target mean r_o, if $H \leq W$.

As mentioned in the preceding paragraph and in Section 2, buffering of pixel values within the box and annulus can significantly reduce the ADGF work requirement. For example, when the ADGF is initialized (i.e., computed at the first target point $\mathbf{y}_1 \in \mathbf{X}$), then the work summarized in Table 1 is incurred. Assuming normal scanning order, if the ADGF is moved across the image, then at the next target point $\mathbf{y}_2 \in \mathbf{X}$, only *min*(2H,2W) addition operations would be performed on the value $r_o(\mathbf{y}_1)$ to obtain $r_o(\mathbf{y}_2)$ with the additional work of one multiplication. This situation is shown schematically in Figure 2 for the more common case of row buffering, where $H \leq W$. The resulting computational efficiency in terms of additions is given by

$$\eta_{rt} = HW / \mathit{min}(2H,2W) .$$

Similarly, if buffering is used during computation of the background mean, then there will be *min*(3H+2e,3W+2e) values replaced instead of 3HW + e(H+W) values in the case where buffering is absent. If $H \leq W$, then row buffering is employed and 6H + 4e additions are required instead of 3HW + e(H+W) additions, as shown in Figure 3.

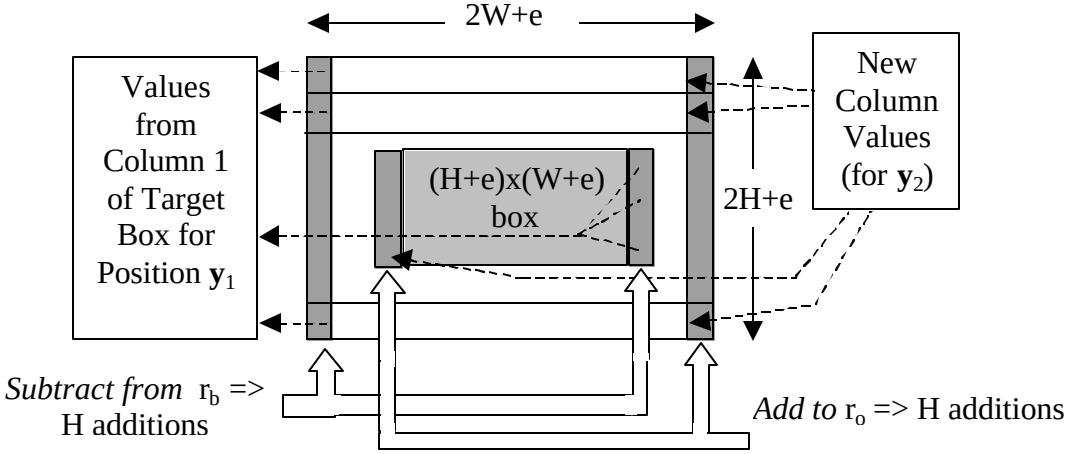


Figure 3. Buffering scheme for efficient computation of background mean r_b, if $H \leq W$.

Thus far, it appears that the DGF can be rendered more efficient by the preceding buffering technique. Unfortunately, when the block standard deviation s_b is considered, the improved ADGF efficiency becomes data-dependent. In particular, consider the differencing operation in the root-mean-sum-of-squared-differences that comprises the standard deviation computation.

This subtraction operation has as its diminuend the background mean r_b. Thus, if r_b changes at $\mathbf{y}_2$ with respect to $\mathbf{y}_1$, then the squared differences associated with the background annulus, whose domain is given by

$$D_b(\mathbf{y}_2) = \mathbf{W}(\mathbf{y}_2) \setminus \mathbf{V}(\mathbf{y}_2)) ,$$

must each be recomputed. This implies recomputation of the squaring (multiplication) operations, as well as the sum of squares. The work requirement of this recomputation is given in the entry of Table 1 specific to s_b.

However, if $r_b(\mathbf{y}_2) = r_b(\mathbf{y}_1)$, then a simpler case occurs. Namely, the squared differences computed from pixels with coordinates in $D_b(\mathbf{y}_1) \setminus D_b(\mathbf{y}_2)$ are subtracted from the quantity $N_b\sigma_b^2$, which is the population-weighted variance of the background annulus. Then, the squared differences computed from pixels with coordinates in $D_b(\mathbf{y}_2) \setminus D_b(\mathbf{y}_1)$ are added to the quantity $N_b\sigma_b^2$, to introduce the new variance information specific to the background at $\mathbf{y}_2$ that is not included in the background at $\mathbf{y}_1$.

Given a DGF target point $\mathbf{y}_i$ and the next target point $\mathbf{y}_{i+1}$ in normal scanning order, where $1 \le i \le |\mathbf{X}| - 1$, the preceding technique is expressed algorithmically as follows:

```
if r_b(y_{i+1}) = r_b(y_i) then
{ sumsq = N_b σ_b^2(y_i) ;
  for each x in D_b(y_i) \ D_b(y_{i+1}) do:
    sumsq = sumsq – (a(x) - r_b(y_{i+1}))^2 ;
  for each x in D_b(y_{i+1}) \ D_b(y_i) do:
    sumsq = sumsq + (a(x) - r_b(y_{i+1}))^2 ;
  σ_b(y_{i+1}) = sqrt( sumsq / N_b ) }
else compute σ_b as usual .
```

If $r_b(\mathbf{y}_{i+1}) = r_b(\mathbf{y}_i)$, and we denote $D_i = D_b(\mathbf{y}_i) \setminus D_b(\mathbf{y}_{i+1})$ and $D_{i+1} = D_b(\mathbf{y}_{i+1}) \setminus D_b(\mathbf{y}_i)$ for purposes of brevity, then it is readily determined by inspection that this algorithm incurs work of $|D_i| + |D_{i+1}|$ + 3 multiplications, $2(|D_i| + |D_{i+1}|)$ additions, and one square root. From Figure 2, it is observed that if $H \le W$, then $|D_i| = |D_{i+1}| = 3H + 2e$.

Table 2. Computational budget for ADGF including buffering of pixel values, where work estimates are listed for non-boundary pixels only.

Stage of Computation	**Additions**	**Multiplications**	**Roots**
Target Mean r_o	2H	1	0
Background Mean r_b	6H + 4e	1	0
Background Stdev σ_b	$f_b(12H+8e)+(1-f_b)(6HW+2e(H+W)-1)$	$f_b(6H+4e+3)+(1-f_b)(3HW+e(H+W)+1)$	1
$TIR^2(\mathbf{y})$	1	2	0
Total Work Required	$8H+4e+f_b(12H+8e)+1+(1-f_b)(6HW+2e(H+W)-1)$	$3+f_b(6H+4e+3)+(1-f_b)(3HW+e(H+W)+1)$	1

Denoting f_b as the fraction of boundary domains D_b that satisfy the conditional in the preceding algorithm, the computational cost for computation of the DGF at non-boundary pixels (assumed to be a small fraction of total pixels) is listed in Table 2.

3.2. Error Analysis

Recall from Section 3.1 that the ADGF output is defined as:

$$\mathrm{TIR}^2 = \left(\frac{\bar{r}_o - \bar{r}_b}{\sigma_b} \right)^2 .$$

In References 7 and 8 we showed that the dataflow graph of an algorithm *A* is isomorphic to the dataflow graph of its error function ε_A. For example, Figure 4a (b) illustrates the dataflow graph of TIR^2 ($\varepsilon_{\mathrm{TIR}^2}$).

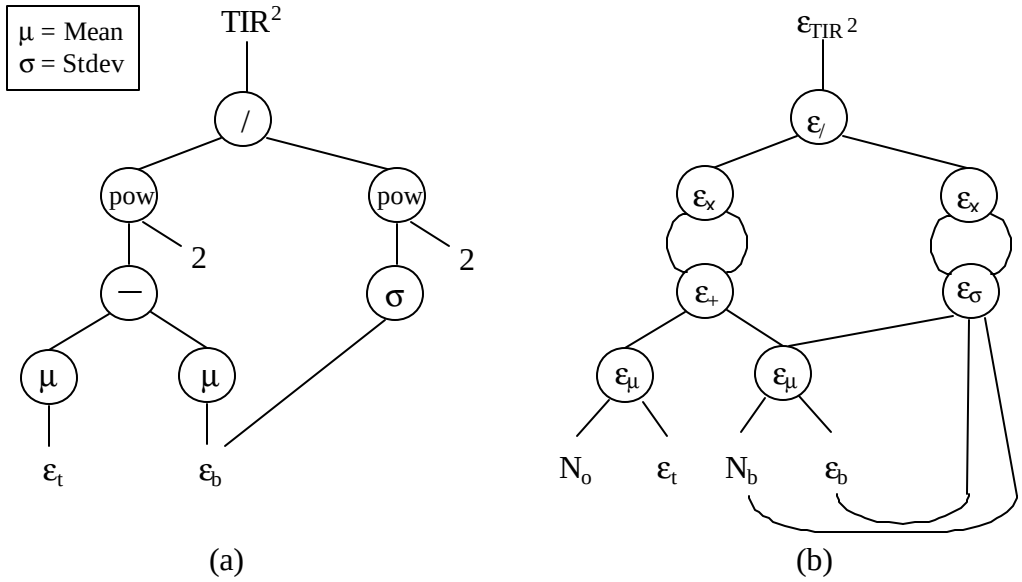


Figure 4. Error function of the ADGF (a) Algorithm dataflow graph, and (b) Error function dataflow graph. The results are TIR^2 and $\varepsilon_{\mathrm{TIR}}2$.

The preceding error function derivation produces a type of error propagation model called *forward error analysis*, which estimates the worst-case error produced by a computation [16]. Here, ε_x and ε_+ respectively denote the forward error propagation functions for multiplication and addition, which are well known. Also, ε_t and ε_b denote the target and background error descriptors, which are discussed below.

In contrast, ε_μ and ε_σ, which denote the error functions of the arithmetic mean μ and standard deviation σ, are less well known, but are discussed in References 7 and 8. In summary the error of the arithmetic mean is the sum of the errors of the terms that are averaged to form the mean. In the ATR imagery under consideration in this study, such errors would be described by a distribution that itself has a mean μ'. In actual practice, when the number of samples N_o or N_b is not small (i.e., $N_o, N_b \geq 10$), the assumption $\varepsilon_\mu = \mu$' holds. Since σ of a constant-valued test target or background region can be approximated by the mean image noise ν, we have found in previous research that it is reasonable to assume that $\varepsilon_\sigma = \nu$.

From Figure 4, we can express the ADGF output error in functional notation as:

$$\varepsilon_{\mathrm{TIR}}2 = \varepsilon_/(\varepsilon_x(\varepsilon_+(\varepsilon_\mu(N_o,\varepsilon_t),\varepsilon_\mu(N_b,\varepsilon_b))), \varepsilon_x(\varepsilon_\sigma(\varepsilon_\mu(N_b,\varepsilon_b),\varepsilon_b))) .$$

In Reference 5, we showed that this expression can be reduced to:

$$\varepsilon_{\mathrm{TIR}}2 = 4(N_o\varepsilon_t + N_b\varepsilon_b + \varepsilon_\sigma)$$

which can be approximated by

$$\varepsilon_{\mathrm{TIR}}2 = 4(N_o\varepsilon_t + N_b\varepsilon_b + \nu) .$$

If we assume that ε_σ is negligible compared to the mean target error ε_t, which approximates σ_t at one standard deviation from ε_t, then we have the further approximation:

$$\varepsilon_{\mathrm{ADGF}} = \varepsilon_{\mathrm{TIR}}2 \approx 4(N_o\varepsilon_t + N_b\varepsilon_b) .$$

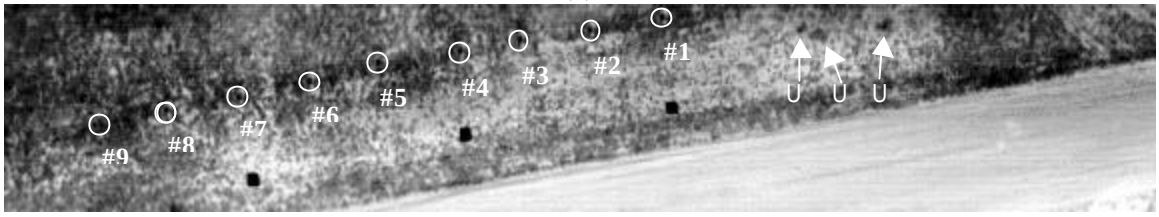
Given the conservative estimates $\sigma_b \approx 1.5\sigma_t$ and $N_b \approx 10N_t$ obtained from analysis of the test imagery (per Section 4), the preceding equation predicts that the ADGF point-to-point response error is influenced by a ratio of 15:1 times background noise or error per unit of target noise or error. As a result, although $\varepsilon_{TIR}2$ appears to scale linearly with background noise ε_b, the ADGF output varies more widely when the background is noisy or highly textured, as shown in Section 5.

4. Analysis of Minefield Imagery

FTI provided UF with a JPEG-encoded version of a high-resolution strip image (1152x209 pixels) that depicts a buried minefield imaged from an airborne sensor platform, as shown in Figure 5a. This image contains nine visually truthed targets deployed along an approximately linear track ranging from the upper center down to the left-center edge of the image, as diagrammed in Figure 5b. A tenth target might exist to the left and downward of Target #9, but cannot be discerned due to the image noise level.

Also identified in Figure 5b are unidentified clutter, denoted by uppercase U, primarily located in the upper right center portion of the image. Observe the dark streak that runs approximately coincident with the mines, which is likely due to grass being trampled during mine deployment. In discussions with Frontier Technology and the sponsor, this observation as well as target identification and location was qualitatively supported by the Army POC. However, ground truth data in the form of centroid coordinates and target dimensions was not provided to the UF research team.

(a)



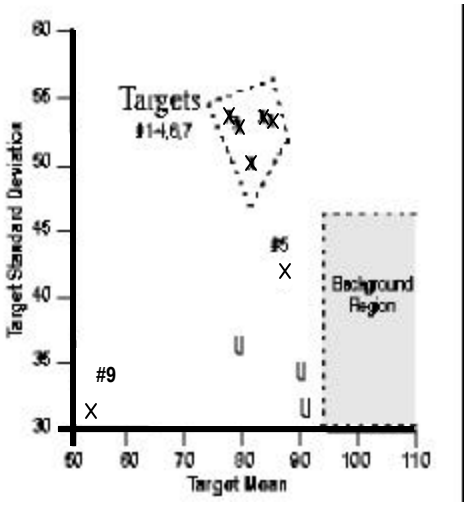
(b)

Figure 5. Minefield imagery: (a) source image, (b) image with candidate targets circled.

After identifying targets, we attempted to separate targets from target-like objects in or around visually identified candidate target positions. This was done by isolating the statistics of target areas of interest with some background included. Such measurements, which are summarized in Figure 6, indicate that a majority of the targets are well separated from the background on the basis of standard deviation, while the use of both mean and standard deviation tends to separate all targets from the background.

Target	Mean	Stdev	N
1	77.3	53.5	156
2	78.6	53.2	147
3	79.9	52.7	151
4	82.8	53.5	133
5	85.4	40.2	129
6	84.0	53.2	156
7	81.4	49.8	180
8	80.7	52.1	164
9	52.8	31.5	180

(a)



(b)

Figure 6. Statistics of candidate targets (x) and unknown objects (U):
(a) entabulated target statistics, and (b) feature plot of mean and standard deviation.

The effect of target and background statistics on the performance of the ADGF was measured using AOI sizes ranging from 300 to 1000 pixels, with target means ranging from 95 to 138 out of 255 greylevels and standard deviation ranging from 30.5 to 47. A portion of this region is denoted by the shaded box in Figure 6b. Given that the target mean within the dashed trapezoidal region in Figure 6b ranges from 75 to 85, and since the ADGF response is given by

$$\mathrm{TIR}^2 = \left(\frac{\bar{r}_o - \bar{r}_b}{\sigma_b} \right)^2 ,$$

Table 3. Example cases of target-background (TB) and object-background (OB) interaction in the context of the ADGF applied to the image of Figure 5a.

Case	Target Mean	Background Mean	Background Stdev	TIR^2
TB0	75/255	138/255	30.5/255	**14.5**
TB1	75/255	95/255	30.5/255	5.49
TB2	85/255	95/255	30.5/255	2.74
TB3	85/255	95/255	47/255	**1.15**
OB0	53/255	95/255	47/255	**4.84**
OB1	83/255	95/255	47/255	1.38
OB2	83/255	95/255	30.5/255	**3.29**

we have that TIR^2 for target and measured background intensities ranges from 1.15 to 14.5, as shown in Table 3. In this table, which is derived from the data of Figure 6, the bounds of the ADGF output range over 1.1 orders of magnitude ($1.1006 = \log_{10}(14.5/1.15)$) for target-background contrast. The unidentified objects that resemble targets occur within the limits of the ADGF's performance range shown in Table 3. This phenomenon could result in false alarms in the absence of appropriate pre- or post-filtering. In Section 3, we showed analytically that background standard deviation can be equated to noise within the context of a simplifying

assumption. In Section 5, we further analyze the adverse effect of image noise on ADGF output, showing that Pd varies inversely with noise.

5. Results of ADGF Applied to Minefield Imagery

The Adaptive Double-Gated Filter was first applied to the minefield image of Figure 5a. Preliminary optimization of the ADGF involved the following steps:

1. Apply ADGF to source image **a**, varying H from 5 to 15 pixels, W from 8 to 20 pixels, and e from 2 to 10 pixels. (The parameters H, W, and e were discussed in Section 2.) Threshold the ADGF output over interval $[T_1,T_2]$ to yield image **b**.
2. Apply a morphological operation called the *rolling ball filter* (RBF, [17]) to the output **b** of the ADGF, where an elliptical template **s** having axes equal to H and W is employed. The output **c** of the RBF is the residue of the opening of **b** using structuring element **s**.

Step 1 detects objects of approximate size HxW pixels that have statistics different from the background surrounding each object. Step 2 is designed to reject false alarms caused by sensor noise or clutter objects such as footprints, clumps of grass or small vegetation, locally packed earth, etc. Optimization iterates Steps 1 and 2 to maximize Ph and reduce Rfa.

A costly but useful alternative to Step 2 involves (a) labeling of all connected components, (b) determining the size of all components from their bounding box, or by fitting of an ellipse to the component, and (c) rejecting all components larger than 1.2H x 1.2W and smaller than 0.3H x 0.3W. The values 1.2 and 0.3 were arrived at in previous studies by examining size of target and false alarm components after thresholding of the DGF. This method will reject candidate targets outside the specified size range, but will not differentiate between mines that are close to each other and, due to point-spread effects, merge to form a larger component.

5.1. Performance Analysis

Preliminary optimization showed that maximum probability of hits (Ph = 1.0) for the ADGF applied to the source image of Figure 5a was found for H = 11, W = 8, e = 5, T_1 = 0.55/1.0, and T_2 = 0.7/1.0. The output image **c** is shown in Figure 6. The total number of target components detected is nine, out of a possible nine targets, which yields unitary Ph.

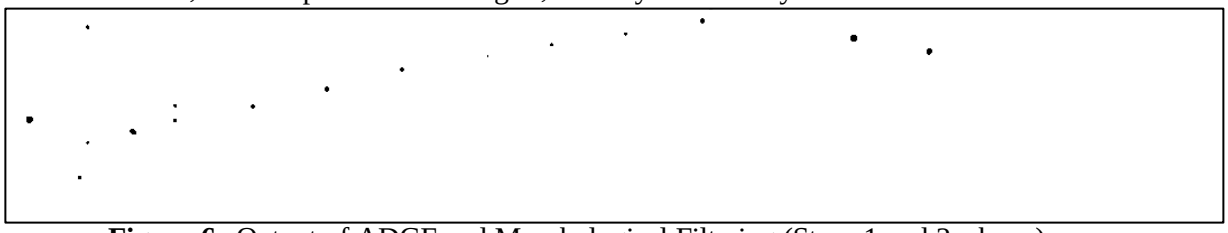


Figure 6. Output of ADGF and Morphological Filtering (Steps 1 and 2, above).

There are nine false alarms in this preliminary optimization result, which include (a) three noise effects in the left-hand side of the image, (b) two apparent foot-tracks – one directly above Target #8 (per Figure 5b), and one to the left of Target #9, which might be another mine; as well as (c) two of the unidentified targets in the right-hand side of the image (marked "U" in Figure 5b). With the exception of the foot track above Target #8, the remaining false alarms can be eliminated by the minefield structure algorithm described in Section 7.

5.2. Noise Analysis.

As discussed in Reference 5, the ADGF is sensitive to noise, as demonstrated in the results of Section 5.1 by the presence of three false alarms attributed to noise. The range of noise variance over which the ADGF performed acceptably was determined from commonly-accepted metrics for occurence of hits and false alarms, respectively expressed in terms of fraction of pixels on- and off-target, per Table 4, as given below.

Since comprehensive performance testing over the entire source image was computationally prohibitive, the source image was partitioned per Figure 7. A suspected target in the right-hand

portion of Figure 7h was included for completeness. Truthing images were obtained via manual tracing using GNU *gimp* software, from visual inspec-

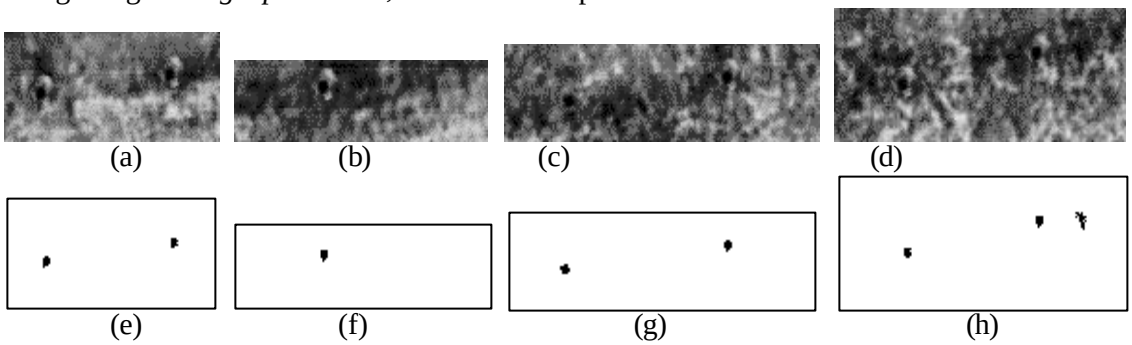


Figure 7. Partitioning of source image into four disjoint subimages (a-d) with visually located truthing data entered in images e) through h).

tion of the source image (Figure 5a) and visually identified targets (Figure 5b). As such, the truthing information is not absolute, but represents a collection of reference data that can be used to determine the noise sensitivity of the ADGF using the parameters listed in Table 4. For example, if **d** denotes the ADGF output and **c** denotes the corresponding truthing image, then

$$N_{tp} = \Sigma\, \mathbf{c} \quad \text{and} \quad N_{tpd} = \Sigma\, (\mathbf{c} * \mathbf{d}) \ ,$$
$$N_{fpd} = \Sigma\, (\mathbf{d} - \mathbf{c}) \quad \text{and} \quad N = \mathit{card}(\mathit{domain}(\mathbf{c})) \ .$$

Table 4. Measures of ATR filter performance employed in ADGF noise analysis.

Measure	Formulation	Description
Fh	N_{tpd} / N_{tp}	Fraction of target pixels correctly detected
Ffa	N_{fpd} / N	Fraction of image pixels that are false alarms
Ph	N_{tcd} / N_{tc}	Fraction of target components detected
Nfa	N_{fcd}	Number of false alarms per image

where the variables are defined as:

N_{tpd} = Number of detected target pixels N_{tp} = Number of actual target pixels
N_{fpd} = Number of false alarm pixels detected N = Number of image pixels
N_{tcd} = Number of target components detected N_{tc} = Number of target components

Additive (Gaussian) and multiplicative (speckle) noise were introduced into each of the images shown in Figures 7a-d via Matlab software. The ADGF was then applied to each image without morphological postprocessing, to obtain a worst-case depiction of noise effects on the ADGF only. Results are shown in Tables 5-8.

Table 5 shows several settings of the ADGF to illustrate the sensitivity of Fh and Ffa to ADGF parameters H, W, and e in the cases studied. In the absence of sufficient resources and GFI imagery to support a more broadly-based analysis of ADGF performance, we conjecture that this sensitivity is due to the relatively narrow point-spread effects observed in the GFI imagery (Figure 5a). However, this could also be due to effects of the JPEG algorithm used to compress the source image before it was provided to FTI and UF. Clearly, further research is required to differentiate these phenomena and their effects on the ADGF performance.

Table 5. ADGF preliminary noise analysis: Baseline results without noise introduced..

Image	(H,W,e)	Fh	Ffa
Fig 7a	(13,10,5)	0.2059	0.0122
Fig 7a	(11,10,5)	0.3824	0.0256
Fig 7a	(11,8,5)	0.3824	0.0203
Fig 7b	(11,10,5)	0.0526	0.0229
Fig 7b	(11,6,5)	0.0526	0.0230
Fig 7b	(11,8,5)	0.0526	0.0186
Fig 7c	(11,10,5)	0.0286	0.0207
Fig 7c	(11,10,3)	0.0286	0.0158
Fig 7c	(11,8,5)	0.2000	0.0213
Fig 7d	(11,8,5)	0.0	0.0099
Fig 7d	(11,10,9)	0.2333	0.0368

In general, the addition of speckle noise decreased Fh and Ffa, due to the decrease in TIR^2 as σ_b increases, as shown in Section 3.2. Similarly, Ph was decreased while Nfa increased. As shown in the representative examples of Table 7, the addition of small amounts of Gaussian noise severely degraded the ADGF performance, due to the additive increase in σ_b with respect to r_o and r_b, which were unchanged. We plan to further investigate this effect in future research, using laboratory (low-noise) imagery with various types and levels of noise.

Table 6. ADGF preliminary noise analysis: Effect of speckle noise with zero mean.

Image	Variance	(H,W,e)	Fh	Ffa
Fig 7a	0.04	(11,10,5)	0.2647	0.0178
Fig 7a	0.04	(11,8,5)	0.2670	0.0167
Fig 7a	0.04	(13,10,5)	0.1765	0.0166
Fig 7a	0.10	(11,10,5)	0.1765	0.0148
Fig 7a	0.10	(11,8,5)	0.2059	0.0119
Fig 7a	0.10	(13,10,5)	0.0294	0.0127
Fig 7a	0.25	(11,10,5)	0.0	0.0072
Fig 7a	0.25	(11,8,5)	0.0	0.0077
Fig 7a	0.25	(13,10,5)	0.0	0.0064
Fig 7b	0.04	(11,10,5)	0.0526	0.0195
Fig 7b	0.04	(11,8,5)	0.0	0.0111
Fig 7b	0.04	(13,10,5)	0.0	0.0138
Fig 7b	0.10	(11,10,5)	0.0	0.0066
Fig 7b	0.10	(11,8,5)	0.0	0.0075
Fig 7b	0.10	(13,10,5)	0.0	0.0055
Fig 7b	0.25	(11,8,5)	0.0	0.0023
Fig 7c	0.04	(11,10,5)	0.0	0.0158
Fig 7c	0.04	(11,10,3)	0.0286	0.0119
Fig 7c	0.04	(11,8,5)	0.1714	0.0186
Fig 7c	0.04	(13,10,5)	0.0	0.0137
Fig 7c	0.10	(11,10,5)	0.0	0.0104
Fig 7c	0.10	(11,10,3)	0.0	0.0072
Fig 7c	0.10	(11,8,5)	0.0571	0.0136
Fig 7c	0.10	(13,10,5)	0.0	0.0102
Fig 7c	0.25	(11,10,5)	0.0	0.0028
Fig 7c	0.25	(11,8,5)	0.0	0.0023
Fig 7c	0.25	(13,10,5)	0.0	0.0020
Fig 7d	0.04	(11,8,5)	0.0	0.0077
Fig 7d	0.04	(11,10,9)	0.1833	0.0336
Fig 7d	0.10	(11,8,5)	0.0	0.0017
Fig 7d	0.10	(11,10,9)	0.2167	0.0309
Fig 7d	0.25	(11,8,5)	0.0	0.0017
Fig 7d	0.25	(11,10,9)	0.1500	0.0079

Table 7. ADGF preliminary noise analysis: Effect of Gaussian noise with zero mean.

Image	Variance	(H,W,e)	Fh	Ffa
Fig 7a	0.0001	(11,8,5)	0.0	0.0087
Fig 7a	0.0002	(11,8,5)	0.0	0.0033

Table 8. Detailed noise analysis: Effect of increasing speckle noise for (H,W,e) = (11,8,5), $T_1 = 0.5$, and $T_2 = 0.7$.

Image	Variance	Fh	Ffa
Fig 7a	0.0	0.3824	0.0203
Fig 7a	0.01	0.3529	0.0189
Fig 7a	0.02	0.3529	0.0144
Fig 7a	0.03	0.3824	0.0164
Fig 7a	0.04	0.2670	0.0167
Fig 7a	0.05	0.2647	0.0153
Fig 7a	0.06	0.2647	0.0139
Fig 7a	0.07	0.2353	0.0153
Fig 7a	0.08	0.2647	0.0144
Fig 7a	0.09	0.3235	0.0127
Fig 7a	0.10	0.2059	0.0119
Fig 7c	0.0	0.2	0.0213
Fig 7c	0.01	0.2	0.0201
Fig 7c	0.02	0.1714	0.0191
Fig 7c	0.03	0.1714	0.0199
Fig 7c	0.04	0.1714	0.0186
Fig 7c	0.05	0.1143	0.0157
Fig 7c	0.06	0.1429	0.0166
Fig 7c	0.07	0.1429	0.0136
Fig 7c	0.08	0.0571	0.0138
Fig 7c	0.09	0.0857	0.0113
Fig 7c	0.10	0.0571	0.0136

The effect of speckle noise was further investigated by increasing the noise variance from 0.01 to 0.1 in steps of 0.01, as shown in Table 8. Despite several anomalies that rendered Fh nonmonotonic with respect to the noise variance, the inverse dependence of Fh on variance that was observed in Table 5 is again manifested in the data of Table 8, as theory predicts.

One might ask why the rate of false alarms does not increase with increased speckle noise variance. This phenomenon is explained by noting that a false alarm is a *detection of a non-target object*, which is not necessarily an artifact of noise. However, as previously noted, σ_b^2 comprises the denominator of the expression for TIR^2. Thus, as σ_b is increased, TIR^2 undergoes inverse square decay, decreasing the total number of detections (targets as well as false targets), thresholds being held constant. As a result, Fh and Ffa both decrease with increasing σ_b. Disambiguation of the non-target detections is partially achieved with the algorithm described in Sections 6 and 7.

The graphs of Figures 8 and 9 illustrate ROC curves for nonzero Fh, plotted on linear axes, where data are taken from the speckle noise results of Tables 6 and 8. In Figure 8 is graphed the

ROC curve as a best-fit line for all cases. Ffa, a crucial performance variable for efficient mine detection and neutralization, is graphed as a function of Fh, the optimized parameter.

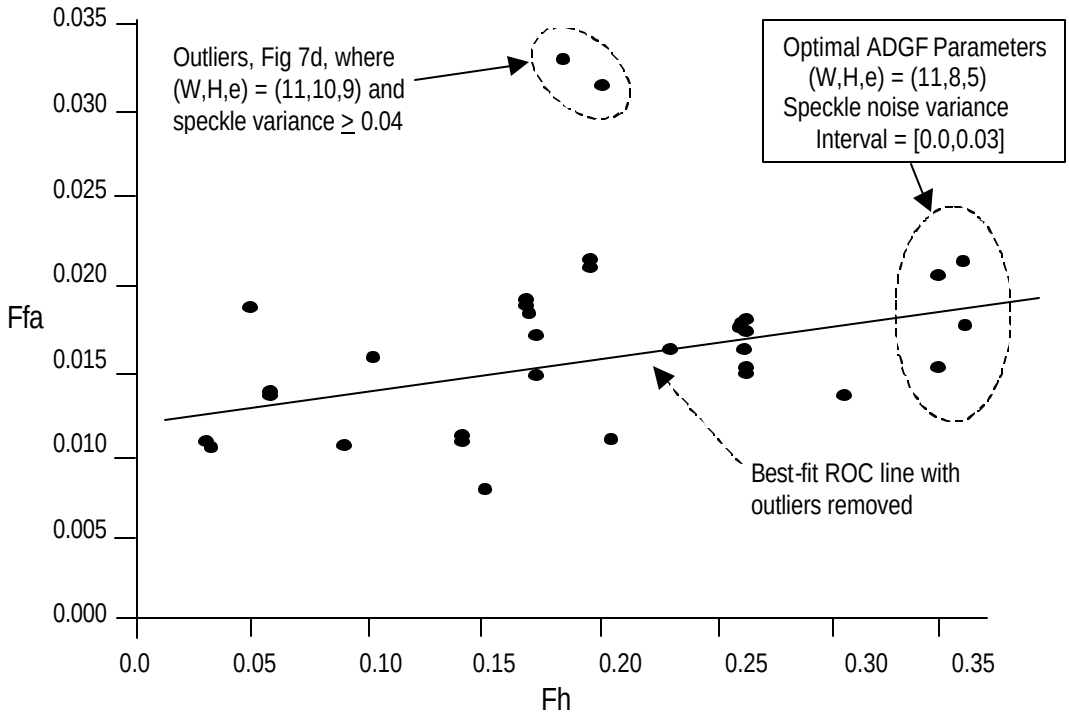


Figure 8. ROC curve of all ADGF test cases where Fh > 0, from Tables 6 and 8.

In contrast, Figure 9 shows only the performance data for Fh > 0 in the optimal ADGF configuration, where (H,W,e) = (11,8,5). Note that noise variance is classified as a) 0.0 through 0.03 (box), b) 0.04 through 0.1 (circle), and c) 0.25 (triangle).

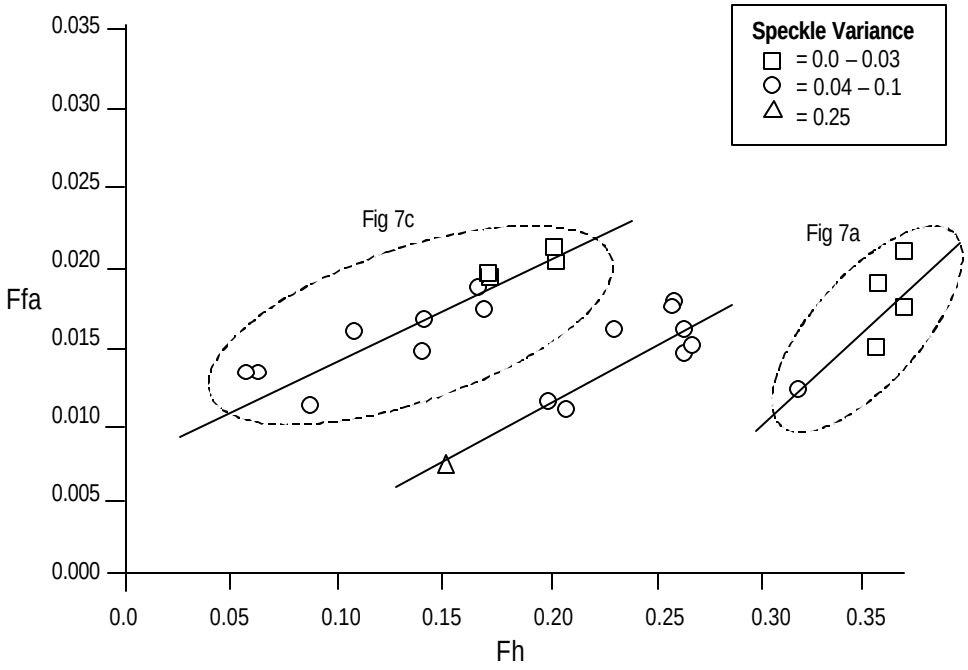


Figure 9. ROC plot of Ffa versus Fh for (W,H,e) = (11,8,5).
Note the clustering of results for Figures 7a and 7c.

Note that the three ROC plots in Figure 9 have several important features. First, Ffa increases with Fh in each of the three cases, as predicted by signal detection theory. Second, Figures 7a and 7c give rise to separate clusters of results. Analysis indicates that this effect is due to less target definition in Figure 7c. This could result from possible greater background noise or target-background similarity. Alternatively, the cluster separation could be due to different compression effects (e.g., distortion due to block boundaries) in the JPEG-compressed image, although such effects are not apparent visually.

It is likewise interesting to note that the within-group range of Ffa values is not significantly different for Figure 7c results versus Figure 7a. This observation is supported in Figure 10 by the similar plots of Ffa versus speckle noise variance for Figures 7a and 7c. Figure 7a has higher Fh, however, which means more pixels on target (and, unsurprisingly, higher Ph). This effect is also supported by the graphs in Figure 10 (Fh versus noise variance).

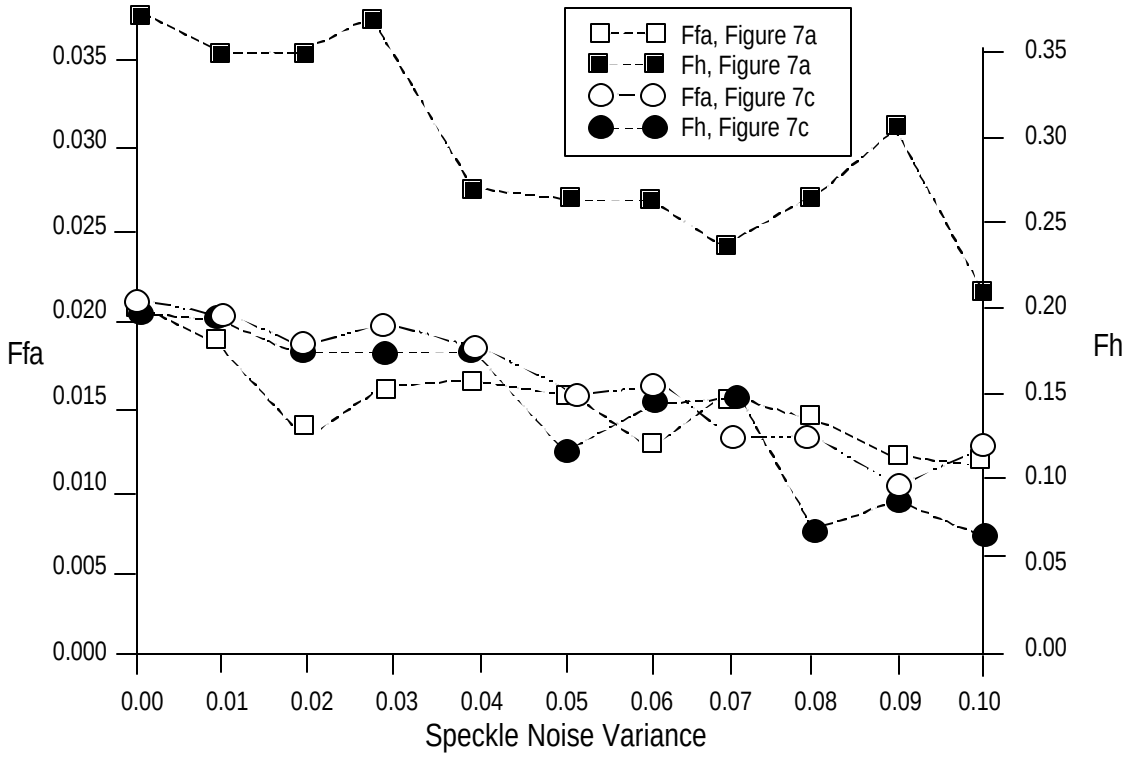


Figure 10. Fh and Ffa as a function of speckle noise variance ranging from 0.0 to 0.1, where (H,W,e) = (11,8,5), and images from Figures 7a and 7c are employed.

Due to the small sample size (nine targets) and the large number of cases where Fh = 0 for ADGF configurations that different from the optimal configuration for the test imagery of (H,W,e) = (11,8,5), it is reasonable to state that the results discussed in this section are case-specific. In Section 8, we propose future work using multiple images at different optical wavelengths (near-ultraviolet, visible, near-infrared, and infrared), to determine relationships of mine signatures with Fh and Ffa for a variety of truthed targets under different field conditions. Such research would better facilitate understanding of the ADGF's behavior under a more realistic variety of field conditions.

6. Minefield Structure Determination

As shown in Section 5, application of the ADGF to minefield imagery occasionally results in undetected targets (low Ph) and spurious detection of non-target objects (e.g., Rfa increases with Ph, per Figure 9). Although morphological postprocessing is well established as a technique for reducing the number of false alarms, we have found that contextual information also facilitates the process of candidate target disambiguation.

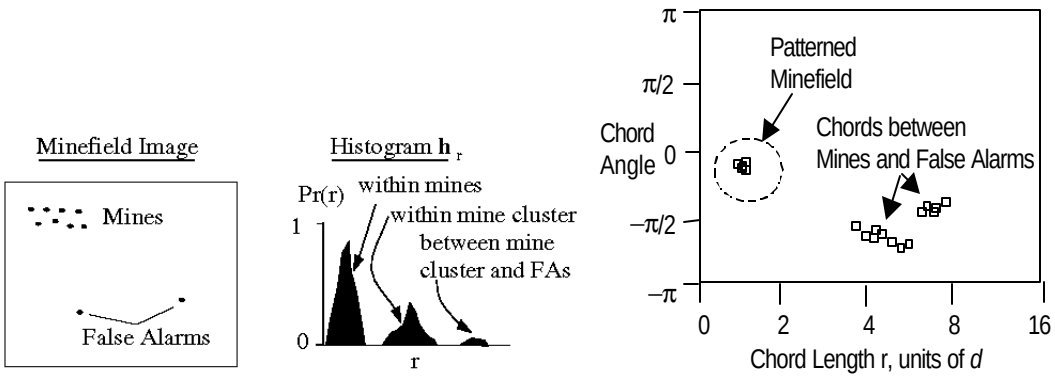
For example, large land mines (e.g., 12 to 15 inches in diameter) are typically emplaced according to quasilinear (e.g., vehicle-deployed targets) or pseudorandom quasi-circular or quasi-parabolic patterns (e.g., scatterable targets). Since the sponsor requested that we analyze linearly emplaced mines in the GFI imagery provided for this study, we have enhanced previous research described in Section 6.1 to include an improved mine detection algorithm for linear tracks, as discussed and analyzed in Section 6.2. In Section 6.3, we show how this enhanced algorithm is applied to the ADGF output presented in Section 5.

6.1. Previous Research

In previous DoD-funded research, UF developed minefield detection and characterization algorithms based on local density (detection step) and chord transforms (characterization step). Local density-based detection uses a pyramidal data structure to encode composite minefield imagery (e.g., mosaicked for large area coverage), given maps of target presence or probability. This establishes the presence, size, and probable spatial location of mine clusters. Assuming that targets are represented as points, the chord transform computes the slope of a line between any two target points. The resulting assembly of lines can be filtered at multiple spatial scales to determine minefield structure.

For example, given a patterned minefield such as that illustrated in Figure 11a, the corresponding chord length transform, illustrated schematically in Figure 11b, would depict elements of field structure at spatial scales corresponding to the von Neumann or Moore neighborhood of each mine. However, larger neighborhoods exhibit a more diverse chord structure, comprised of many different slope angles, as shown in Figure 11c. This observation provides a basis for distinguishing between patterned and non-patterned (i.e., pseudorandom) minefields. In practice, we have found that pseudorandom patterns have chordal distributions that are pseudorandom at spatial scales larger than the mean inter-mine spacing d, whereas regularly patterned minefields have regular chordal structure up to approximately $2.5d$.

Since the chord transform can be computed within a moving window, it is also possible to account for space-variant structure, which is comprised of both patterned and pseudorandom (scattered) mine emplacements. Given the fact that large mines (e.g., M-19s) are usually distributed from a vehicle moving on an approximately linear track, and that smaller (antipersonnel) mines are typically scattered (e.g., from artillery or airborne platforms), it is possible to achieve increased efficiency by adjusting spectral filter tolerances and data fusion constraints to select different mine types and thus "seed" the minefield detection process with an estimate of probable field structure.



(a) (b) (c)

Figure 11. Notional diagram of minefield detection using a chord transform: (a) Regularly patterned minefield with false alarms, (b) chord-length transform of a), (c)chord length and orientation transform of a).

6.2. Minefield Structure Algorithm

The chord-based field detection algorithm employed in this Phase-I study is expressed in pseudocode, as follows:

Step 1. Seed the algorithm, preferably with three approximately colinear targets, to construct two line segments $L_1 = \langle \mathbf{p}_{01}, \mathbf{p}_{11} \rangle$ and $L_2 = \langle \mathbf{p}_{11}, \mathbf{p}_{21} \rangle$ with respective orientation θ_1, θ_2 where $\theta_1 = tan^{-1}[p_2(\mathbf{p}_{11} - \mathbf{p}_{01}) / p_1(\mathbf{p}_{11} - \mathbf{p}_{01})]$ and length r_1, r_2 where $r_1 = \| \mathbf{p}_{11} - \mathbf{p}_{01} \|$, with p_k denoting projection to the k-th coordinate and ($\| \|$) denoting the norm.

Step 2. Move along L_2, away from L_1, until the next detected object is encountered (e.g., at $\mathbf{p}_{31}$). If this object forms an angle θ_3 with L_2 that is within a prespecified angular range $[\theta - \Delta\theta, \theta + \Delta\theta]$ and has length r_3 within a prespecified range $[r - \Delta r, r + \Delta r]$, then mark $\mathbf{p}_{31}$ as a *probable* target location.

If $\theta - 2\Delta\theta \leq \theta_3 < \theta - \Delta\theta$ and $\theta + \Delta\theta < \theta_3 \leq \theta + 2\Delta\theta$, with r in $[r - 2\Delta r, r + 2\Delta r]$, then $\mathbf{p}_{31}$ is marked as a *possible* target location. Conversely, $\mathbf{p}_{31}$ can also be marked as a possible target location if q is in $[\theta - 2\Delta\theta, \theta + 2\Delta\theta]$ and $r - 2\Delta r \leq r_3 < r - \Delta r$ with $r + \Delta r < r_3 \leq r + 2\Delta r$.

Otherwise, $\mathbf{p}_{31}$ denotes the location of a false alarm.

Step 3. Repeat Step 2 until all objects are labeled. It may be necessary to return to Step 1, reversing the scanning order along an apparent line of mines if the algorithm is seeded with mines that are in the middle of a line.

Complexity Analysis. Let M mines be, on average, in a circular neighborhood N of diameter $4\Delta r$, to facilitate the search process described in Step 2, above. Let N be examined for candidate targets for each of N_t candidate targets in a source image. The average work performed by the preceding algorithm is one invocation of Step 1 and $\mathbf{O}(MN_t)$ invocations of Step 2. Determination of the angle θ requires two additions, one multiplication, and two comparisons to determine if $tan(\theta)$ is within range. Determination of the distance r requires three additions, two multiplications, and two comparisons to ascertain if r^2 is within range. Thus, the mean computational work is given by:

$$\underline{W}(M,N_t) = MN_t(5 \text{ additions} + 3 \text{ multiplications} + 4 \text{ comparisons}) .$$

6.3. Test Results

An example implementation of the chord-based minefield detection algorithm given in Section 6.2 is illustrated in Figure 12, which is based on the DGF output shown in Figure 5b. This output is labeled in Figure 12, for purposes of clarity (i.e., nine targets T1-T9 and seven false alarms F1-F7). Compared with the discussion in Section 5, one false alarm was here relabeled as a candidate target (#10 in Figure 12). Table 9 lists the centroid coordinates of each target and false alarm, in the context of a 1090x193-pixel version of Figure 5a (minified with respect to the source image, for clarity). Assume that T2, T3, and T4 are applied to seeding the minefield detection algorithm, which is assumed to operate under the assumption of a quasilinear minefield configuration, as discussed previously.

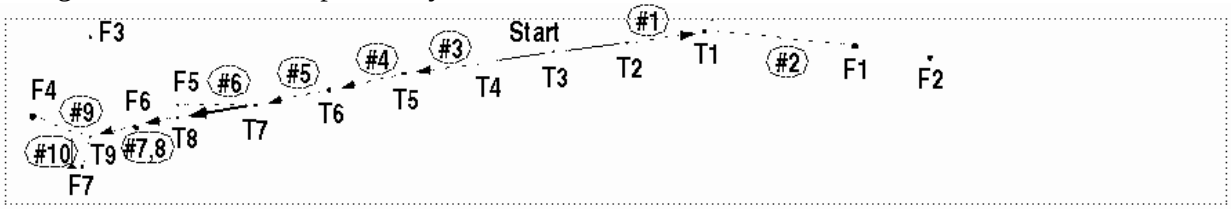


Figure 12. Chord-based minefield detection applied to the DGF output of Figure 6, whose boundary is denoted by a dotted line.

As mentioned in Step 3 of the preceding algorithm, since we chose to seed the algorithm with targets T2-T4, the algorithm would be applied in two different directions. First, go from T2 to T1, then test F1 and F2 as possible endpoints of a new line segment originating at T1.

Table 9. Centroid coordinates for target and selected false target objects in Figure 12.

T1:	(623, 12)	T2:	(554, 23)	T3:	(489, 32)	T4:	(431, 42)
T5:	(355, 55)	T6:	(288, 72)	T7:	(223, 87)	T8:	(152,101)
T9:	(75,120)	F1:	(757, 27)	F2:	(825, 38)	F3:	(74, 17)
F4:	(23, 99)	F5:	(152, 87)	F6:	(116,110)	F7:	(67,151)

Second, go from T4 to T5, T6, and so forth, to complete the chain of detected mines. In the following explanation, the numbered steps (e.g., #1), correspond to the numbered arrows or line segments in Figure 6. Without loss of generality, for simplicity and consistency, all angles in this example are computed positive counter clockwise with respect to the horizontal. The example computation proceeds as follows:

Seed: Start with L_1 = <T2,T3>, L_2 = <T3,T4>, where $\theta_1 = 7.88^o$, $r_1 = 65.6$ pixels, and $\theta_2 = 8.36^o$, $r_2 = 68.7$ pixels. Choose the angular (length) range at $8^o \pm 2^o$ (67 ± 16 pixels), to effect an error tolerance of approximately 25 percent.

The following are numbered iterations of Step 2:

#1: Since T2 is near the top of the image, investigate any remaining targets near the upper boundary by first applying the preceding algorithm in reverse, that is, moving from L_2 to L_1 and thence to T1. Here, $q = 9.05^o$ and r = 69.9 meet the criteria in Step 1 of the algorithm, so T1 is marked as a probable target.

#2: The false targets F1 and F2 meet neither the prespecified angle nor distance criterion with respect to T1 (e.g., <F1,T1> has angle -6.38^o, which is outside the interval $[\theta - 2\Delta\theta, \theta + 2\Delta\theta]$ and length 134.84, which is outside $[r - 2\Delta r, r + 2\Delta r]$, and similarly for F2). Thus, F1 and F2 are labeled false alarms.

#3: Now applying the algorithm as stated previously (going from L_1 to L_2 and so forth), we encounter T5, where <T4,T5> has angle 9.7^o and length 77.1, so T5 is labeled a probable target.

#4,5: As in #3, T6 and T7 are labeled probable targets.

#6: This step examines two targets F5 (T8) with $\theta = 0^o$ (11.2^o) and r = 71 (72.3) pixels. Although both F5 and T8 have r within prespecified bounds for a probable target, θ for F5 is out of bounds, so T8 is labeled a probable target and F5 is labeled a false alarm.

#7: F7 is labeled a false alarm because r = 37.1 pixels and $\theta = 13.7^o$.

#8: T9 is erroneously labeled a false alarm, but not a probable target, since r = 79.3 pixels (within bounds) but $q = 13.8^o$. This error could be remedied by increasing $\Delta\theta$ from 2 degrees to 3 degrees, without loss of accuracy in the other detections.

#9,10: F4 and F7 have far too large an angle with respect to T9, and are thus labelled false alarms.

The accuracy of the preceding minefield detection algorithm depends on four factors: (1) the spatial accuracy of DGF output, (2) image resolution, (3) local linearity of the minefield pattern being traced, and (3) values of $\Delta\theta$ and Δr. We plan to analyze spatial error of the ADGF in the proposed Phase-II study. Image resolution imposes well-established limits on length and orientation error of line segments, which are extensively quantified in the computer graphics literature. The linearity of the minefield pattern is an underlying assumption that can be tested using techniques presented in Reference 11.

7. Statistical Vector Quantization (SVQ) Methods for Mine Detection

SVQ is a pattern recognition process that is both simple and yet very powerful. The formulation can be implemented at low computational cost via FTI's TNE™ algorithm.. Several considerations lead to the formulation of SVQ, which are:

1. Imagery of real objects (e.g., landmines) is usually corrupted somewhat by obscurations (e.g., grass, rocks), collection system artifacts, etc. Additionally, objects in the field rarely match predetermined models exactly. Thus, the traditional recognition process of simply vectorizing pixel patches in a test image, then comparing these vectors with stored exemplar images is likely to be ineffective.
2. However, smaller blocks of pixels taken from mines and other man-made objects tend to have different statistics and appearance than similar pixel blocks taken from natural objects such as grass or earth. Certainly, no single small pixel patch is an adequate basis for target detection, but one might reasonably ascertain whether an image patch is more grass-like, more mine-like, or neither.
3. Groups of contiguous pixel patches, each with a distinct mine-like signature, would indicate the presence of a mine-like object.

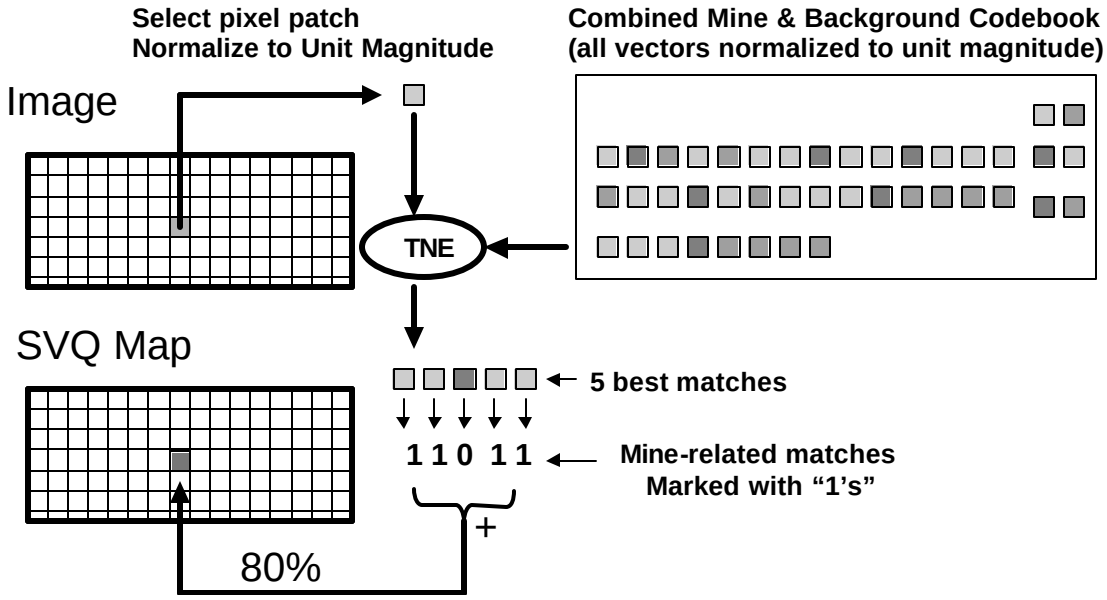


Figure 13: Notional diagram of the SVQ process.

In a sense, SVQ can be thought of as a bloodhound. We train the SVQ process with image partitions from training objects (e.g., we help the bloodhound get the scent). Concurrently, we train the classifier with respect to the anticipated background. We then allow the TNE™ process to compare each pixel patch in an image of interest with combined codebooks that are derived from the training objects and the background. This process is illustrated in Figure 13.

Figure 14 shows an example. Here we used the left-hand 66 percent of the minefield image shown in Figure 5. To train the SVQ process, we cropped several pixel patches containing mine images from the original image. Two of these images were replicated several times to get more pixel phase shift variety. We also cropped two larger portions of the image background. The codebook formation software stepped through these images and formed 8x8-pixel blocks for use in the target or background codebooks, as appropriate. These component codebooks are

combined into a single codebook, where the origin of each pixel patch or codebook vector is stored, and forms a basis for the tables that support function of the TNE™ algorithm.

The prototype SVQ processor developed in Phase I operates on the example image shown in Figure 14, as follows:

1. The image is tessellated into fixed-size tiles.
2. Each tile is normalized to unitary magnitude, and then compared with each exemplar in the combined codebook to determine the number of common corresponding pixel values, within a preset tolerance. Each comparison between a test tile and a codebook exemplar is assigned a matching score *s*.
3. SVC then sorts the codebook vectors based on their level of agreement with the test tile, which is called the number of *pixel agreements*.
4. From the sorted list, the best matching codebook vectors are selected according to an order-statistical criterion (e.g., the highest 0.1 percentile).
5. A measure M is assigned to each image partition (tile) that corresponds to the number of codebook exemplars that were associated with reference (target) data (e.g., mines). A new image is formed that displays M as a pixel value in the output image, at a location corresponding to the tile from which M was derived.

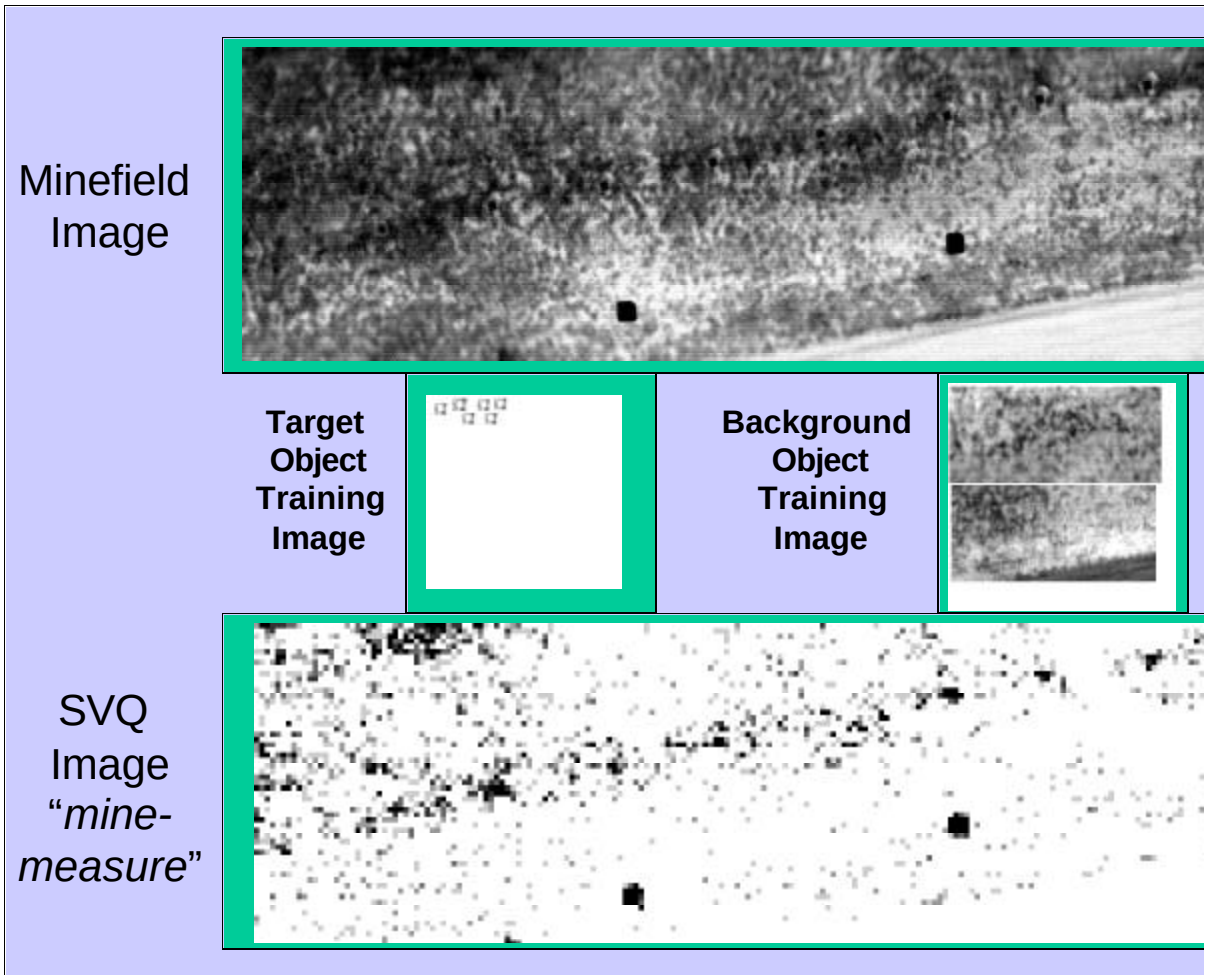


Figure 14. Example of SVQ applied to a minefield image. Note that in addition to the mines themselves, the procedure also highlights the adjacent regions. This results from training on the surrounding regions.

Figure 14 illustrates the power of the SVQ technique. Note that the SVQ output image not only highlights the entire collection of mine targets, but also the surrounding areas of disturbed grass and earth. This is to be expected, since the classifier was trained on the grassy portions as well as the mines.

8. Conclusions and Future Work

Research supporting Frontier Technology, Inc.'s Phase-I SBIR effort has been conducted at University of Florida under sponsorship of US Army (SBIR Topic A99-036), pertaining to detection of surface and buried land mines. At the sponsor's request, UF and FTI's research concentrated on: (a) enhancement of the double-gated filter – a center-surround geometric/statistical detector featured in UF's previous mine detection research, which was extended to perform adaptively in response to varying sensor range-to-target and look angle; (b) application of the resulting Adaptive Double-Gated Filter (ADGF) to GFI digital imagery of a minefield containing large buried mines and numerous clutter objects; (c) performance test and analysis of the ADGF on GFI imagery; (d) determination of the ADGF's noise sensitivity; and (e) development of an algorithm to enhance mine detection software or systems by exploiting the underlying structure of vehicle-deployed minefields such as that depicted in GFI imagery.

In response to the sponsor's advice and direction, UF completed research items a) through e), above, resulting in (1) theory for configuring the ADGF in response to varying sensor parameters, (2) Matlab m-codes for the ADGF, (3) test results specific to applying the ADGF to GFI test imagery, (4) error and noise analysis of the ADGF with supporting test results, and (5) an algorithm for disambiguating minefield structural information to facilitate mine detection given ADGF output, with example results derived from output of the ADGF applied to GFI imagery.

The ADGF was shown to detect all targets in the GFI imagery that could be discerned by humans with the aid of a computer-based image display. We demonstrated that probability of correct target detection was increased, and detection probability of false targets was reduced, by optimizing DGF parameters such as height H and width W of the target area as well as standoff distance e between the target area and background sampling annulus. An optimal value of (H,W,e) = (11,8,5) was found for the GFI imagery. In plots of receiver operating characteristic (ROC), the fraction Ffa of false alarm pixels as a fraction of total image pixels was shown to increase quasilinearly with the fraction of target pixels detected (Fh), as detection theory predicts.

Noise sensitivity of the ADGF was shown to be dependent on background noise, in particular, speckle noise whose variance ν, taken over an interval of [0.0,1.0], exceeded 0.03 . We found that the fraction of target pixels correctly detected (Fh), as well as the fraction of total image pixels that were false alarms (Ffa), declined (in theory) as $1/\nu$, and in practice, decreased quasilinearly as ν increased. Test result sample size was insufficiently large to support the theoretical prediction of $Fh = k/\nu$, where k is a constant. Additionally, the ADGF noise performance results were found to be space-variant over the GFI test imagery, suggesting the need for more and varied test imagery, for example, a larger GFI image dataset in a follow-on (Phase II) effort.

The minefield structure detection and exploitation algorithm performed well on the ADGF results obtained from GFI imagery. All targets were detected, and the rate of false alarms was decreased by a factor of greater than two. As before, more test data are required to determine sensitivity of the algorithm to realistic variation in mine location, effects of realistic camera noise and point-spread effects, and realistic populations and distributions of clutter objects.

Future work has been proposed for Phase-II of this SBIR effort. Proposed research includes support of FTI's technical efforts in (a) enhancement and evaluation of ATR filters that could work together with the ADGF to provide a rich variety of features for target detection or

(possibly) recognition, (b) development of one or more data fusion operators to combine the outputs of the ATR filters, (c) development of more extensive and detailed analysis and test procedures for evaluating the performance of mine detection algorithms and software under a variety of realistic conditions.

9. Bibliography

[1] Bose, P. *Image Restoration via Anisotropic Nonlinear Diffusion and Time-Delay Regularization*, M.S. Thesis, Department of Mathematics, University of Florida (2000).

[2] Key, G., M.S. Schmalz, and F.M. Caimi. "Performance analysis of tabular nearest neighbor encoding algorithm for joint compression and ATR over compressed imagery. 1. Background and theory", *Proceedings SPIE* **3814**:115-126 (1999).

[3] Key, G., M.S. Schmalz, and F.M. Caimi. "Performance analysis of tabular nearest neighbor encoding algorithm for joint compression and ATR over compressed imagery. 2. Results and analysis", *Proceedings SPIE* **3814**:127-142 (1999).

[4] Yang, C.T., W.C. Hu, M.S. Schmalz, and G.X. Ritter. "Center-surround filters for the detection of small targets in cluttered multispectral imagery. 1. Background and filter design", *Proceedings SPIE* **2496**:756-766 (1995).

[5] Schmalz, M.S. , G.X. Ritter, C.T. Yang, and W.C. Hu. "Center-surround filters for the detection of small targets in cluttered multispectral imagery. 2. Analysis of errors and filter performance", *Proceedings SPIE* **2496**:756-766 (1995).

[6] Schmalz, M.S., G.X. Ritter, R. Forsman, C.T. Yang, W.C. Hu, R.A. Porter, G. McTaggart, J.F. Hranicky, and J.F. Davis. "Software requirements and support for image-algebraic analysis, detection, and recognition of small targets", *Proceedings SPIE* **2496**:736-747 (1995).

[7] Schmalz, M.S. "Automated analysis and prediction of accuracy and performance in ATR algorithms. 1. Requirements, theory, and software implementation", *Proceedings SPIE* **3079**:249-260 (1997).

[8] Schmalz, M.S. "Automated analysis and prediction of accuracy and performance in ATR algorithms. 2. Experimental results and analysis of system performance", *Proceedings SPIE* **3079**:261-272 (1997).

[9] Andrews, A.M., V. George, and T.W. Altschuler. "Quantifying performance of mine detectors with fewer than 10,000 targets", *Proceedings SPIE* **3079**:273-292 (1997).

[10] Dougherty, S. and K.W. Bowyer. "Objective evaluation of edge detectors using a formally defined framework", in *Empirical Evaluation Techniques in Computer Vision*, Los Alamalitos, CA: IEEE Computer Society Press (1998).

[11] Schmalz, M.S. and G.X. Ritter. "An introduction to the detection of target field structure with data fusion of density and chord information", *Proceedings SPIE* **2765**:397-408 (1996).

[12] Schmalz, M.S., W.C. Hu, and G.X. Ritter. "Small target detection in multispectral imagery with Cyclic Overlay Processing", *Proceedings SPIE* **2758**:69-81 (1996).

[13] Ritter, G.X., P. Sussner, and J.L. Diaz-de-Leon. "Morphological associative memories", *IEEE Transactions on Neural Networks* **9**(2):281-293 (1998).

[14] Ritter, G.X. and T. Beavers. "Introduction to morphological perceptrons", *Proceedings of the 1999 Artificial Neural Networks in Engineering Conference (ANNIE '99, St. Louis, MO)* **9**:173-178 (1999).

[15] Ritter, G.X. and J.N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*, Second Edition, Boca Raton, FL: CRC Press (2000).

[16] Adams, G.E. *Precision-Based Program Optimization*, Ph.D. Dissertation, Department of Electrical Engineering, Cornell University (2000).

[17] Kneen, M.A. and H.J. Annegarn. "Algorithm for fitting XRF, SEM and PIXE X-ray spectra backgrounds", *Nuclear Instruments & Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms* **109-110**:209-213 (1996).

[18] F. M. Marchak, "Fractal Models in the Visual Perception of Textures and Surfaces in Nature", *Ph.D. Thesis*, Dartmouth College, (1987).

[19] Y. Fisher (editor), *Fractal Image Compression: Theory and Application*, Springer-Verlag, NY, (1995).

[20] A. P. Pentland, "Fractal-Based Description of Natural Scenes", *IEEE Trans. on Patt. Analysis and Machine Intel.*, **6**(6), 661-674, (1984).

[21] N. Sarkar and B. B, Chaudhuri, "An Efficient Differential Box-Counting Approach to Compute Fractal Dimension of Images", *IEEE Trans. Sys. Man, Cybernetics*, **24**(1), (1994).

References Not Cited in Text:

Ritter, G.X. and M.S. Schmalz. "Image-algebraic analysis, detection, and recognition of surface mine targets. 1. Project overview and summary of image algebra", in *Proceedings of the 1993 ONR/CSS Mine Countermeasures Image Analysis Workshop, Panama City, FL 16-17 Nov 1993.*

Ritter, G.X. and M.S. Schmalz. "Image-algebraic analysis, detection, and recognition of surface mine targets. 2. Overview of technical approach, theory, and software development effort", in *Proceedings of the 1993 ONR/CSS Mine Countermeasures Image Analysis Workshop, Panama City, FL 16-17 Nov 1993.*

Ritter, G.X. and M.S. Schmalz. "UF/SMDG – Surface Mine Detection with Differential Multispectral Data Fusion", Final Report for Contract No. N61331-93-C-0048 (1996).

Uttal, W.R. *Visual Form Detection in Three-Dimensional Space*, Hillsdale, NJ: Lawrence Erlbaum (1983).

Appendix Advanced Technology Summary

This appendix summarizes key technological advances that are required to support future research in mine detection. For example, in a Phase-II proposal for a follow-on research and development effort, we have highlighted several areas of research interest that represent an extension of UF and FTI's ATR modeling, simulation, and analysis technologies. Proposed work and supporting technologies are summarized for in-depth analysis of front-end ATR filters, enhancement of FTI's TNE pattern recognition paradigm, and extension of UF's error analysis tools, to be applied to ATR filters and data fusion operators.

Front-End Filter Analysis. A key task in a follow-on effort would be detailed performance and error analysis of gradient or edge detection algorithms such as the Sobel or Kirsch edge detectors, which typically support pixel-level and region-based target detection in other mine detection systems. For example, the Northrop-Grumman (N-G) mine detection algorithm of interest to the sponsor employs a Sobel edge detector, whose error modes were not isolated in previous studies. Without such analysis, it is difficult to accurately determine the effect of Sobel output errors on the data fusion paradigm inherent in the N-G algorithm.

The modeling, simulation, and analysis of ATR filter noise performance represents a low-risk, high-payoff effort that would employ standard methods of error analysis including forward and backward error propagation, sensitivity analysis, and random parameter perturbation

(discussed in Section B.3), to quantify ATR filter accuracy and performance in a variety of noise scenarios. In particular, it is necessary to verify published analyses with tests on in-house GFI minefield imagery perturbed with various noise levels and distributions (e.g., Gaussian, Poisson, speckle, and Lorentzian noise). Of key interest are speckle and Lorentzian noise, due to their involvement in the process of image intensification, which is a component of low-light surveillance imaging.

Extension of TNE Pattern Recognition Paradigm. FTI's TNE paradigm has been analyzed in previous collaborative research between UF and FTI, but lacks a unifying treatment of complexity and error that would enable optimization of TNE parameters for a given ATR scenario. To support more sophisticated use of TNE in mine detection applications, it is necessary to analyze TNE's performance on various types of minefield imagery, including surface and buried mines, to determine bounds on representational error within the limit of TNE's prespecified matching error. UF has proposed to assist FTI in researching and developing means for estimating the distribution of matching errors prior to the application of TNE to a given image test set having known statistics. This would greatly extend the utility of the TNE algorithm in image compression and ATR, and could enable greater understanding of TNE that would lead to further enhancements supporting more efficient operation, for example, on embedded processors in UAVs and UCAV-based imaging scenarios.

If successful, the results of such analysis could support applications such as more accurate image compression, encoding, and ATR over vector-quantized (VQ) imagery. As we have shown in previous research, computation over VQ-compressed imagery can yield a potentially high speedup, especially for applications that are constrained to architectures comprised of one or more embedded processors. This holds particularly for computational systems that operate under space, time, and power constraints (e.g., on small autonomous vehicles such as UAVs, UCAV, and micro-air vehicles).

Enhanced Error Analysis Theory and Software.

In order to assist FTI in analyzing sensitivity of algorithm implementations to systematic perturbations such as camera noise or sensor pointing error, UF must develop error analysis theory and software that has a broader and more detailed scope than found in existing practice. In particular, we have proposed to assist FTI in analyzing *implementational error*, versus error in theoretical expressions or algorithms that embody such theory. Implementational error analysis, which is supported by error propagation theory and models, is necessary due to the wide variety of computational platforms that could be employed in a fielded system. For example, IBM/PC architectures running 32-bit FORTRAN arithmetic libraries, UNIX systems running C or C++ with variable-precision arithmetic, and embedded processors with proprietary or UNIX-like operating systems running limited-precision computation of arithmetic and transcendental operations. The comprehensive error analysis that UF has proposed to develop could be used to assist FTI in measurement of computational error using realistic (GFI) minefield imagery on selected computational platforms (e.g., the PC and UNIX systems discussed previously).

In order to efficiently carry out the detailed implementational error analyses discussed above, UF should first enhance its prototype error profiling software [7,8] to more accurately estimate typical and best-case error. Our early prototype theory and software currently supports worst-case error analysis, but often produces error estimates that are too conservative. Such development would entail moderate risk but high payoff. A small risk would be incurred by the adaptation of existing published theory (from numerous sources) for error propagation in linear and quasi-linear operations, whereas the development of enhanced error estimation theory for nonlinear operations would entail moderate risk. High payoff would be realized by the development of an error analysis system that could support FTI's and UF's future collaborative research in algorithm design, analysis, and implementation for a wide variety of applications.