ARMY RESEARCH LABORATORY

Automatic Target Acquisition of the DEMO III Program

by Sandor Der, Alex Chan, Gary Stolovy, Michael Lander, and Matthew Thielke

ARL-TR-2683 **August 2002**

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-2683 **August 2002**

Automated Target Acquisition for the DEMO III Program

Sandor Der, Alex Chan, Gary Stolovy, Michael Lander, and Matthew Thielke
Sensors and Electron Devices Directorate, ARL

Contents

List of Figures

List of Tables

1. Introduction

This work was performed for the DEMO III Unmanned Ground Vehicle (UGV) program, which is developing UGVs that will assist U.S. Army scouts. The Electro-Optics Infrared (EOIR) Image Processing branch (AMSRL-SE-SE) has been tasked with developing algorithms for acquiring and recognizing targets imaged by the Wescam Forward-Looking Infrared (FLIR) sensor. These images are sent back to the user upon request or when the automatic target recognizer (ATR) indicates a location of interest. The user makes the ultimate decision about whether an object in an image is actually a target. The ATR reduces the bandwidth requirement of the communication link because the imagery can be sent back at reduced resolution, except those regions indicated by the ATR as being possible targets. The algorithms consist of a front-end detector, a clutter rejector, and a recognizer. The next three sections describe these components.

2. The Detection Algorithm

The algorithm described in this report was designed to address a need for a detection algorithm with wide applicability which could serve as a prescreener/detector for a number of applications. While most automatic target detection/recognition (ATD/R) algorithms use much problem-specific knowledge to improve performance, the result is an algorithm that is tailored to specific target types and poses. The approximate range to target is often required, with varying amounts of tolerance. For example, in some scenarios, it is assumed that the range is known to within a meter from a laser range finder or a digital map. In other scenarios, only the range to the center of the field-of-view and the depression angle is known so that a flat earth approximation provides the best estimate. Many algorithms, both model-based and learning-based, either require accurate range information or compensate for inaccurate information by attempting to detect targets at a number of different ranges within the tolerance of the range. Because many such algorithms are quite sensitive to scale, even a modest range tolerance requires that the algorithm iterate through a large number of closely spaced scales, driving up both the computational complexity and the false alarm rate. Algorithms have often used statistical methods [1] or view-based neural networks [2, 3, 4].

The proximate motivation for the development of the scale-insensitive algorithm was to provide a fast prescreener for a robotic application for which no range information was available. The algorithm instead attempted to find targets at all ranges between some reasonable minimum, determined from operational requirements and the maximum effective range of the sensor.

Another motivation was to develop an algorithm that could be applied to a wide variety of image sets and sensor types, which required it to perform consistently on new data, without the severe degradation in performance that commonly occurs with learning algorithms, such as neural networks and principal component analysis (PCA)-based methods, that have been trained on a limited variety of sensor types, terrain types, and environmental conditions. While we recognize

that with a suitable training set, learning algorithms will often perform better than other methods, this typically requires a large and expensive training set, which is sometimes not feasible.

2.1 The Data

The dataset used in training and testing this system was the April 1992 Comanche FLIR collection at Fort Hunter-Ligget, CA. This dataset consists of 1225 images, each 720 by 480 pixels. Each image has a field of view of ~1.75° squared.

Each of the images contains one or two targets in a hilly, wooded background. Ground truth was available, which provided target centroid, range-to-target, target type, target aspect, range-to-center of field-of-view, and the depression angle. The target centroid and range-to-target were used to score the algorithm, as described in the experimental results section, but none of the target-specific information was used in the testing process. The algorithm only assumes that the vertical and horizontal fields of view and the number of pixels horizontally and vertically is known. The only range information used is the operational minimum range and the maximum effective range of the sensor.

2.2 The Features

Each of the features is calculated for every pixel in the image. As more complex features are added in the future, it might become beneficial to calculate some of the features only at those locations for which the other feature values are high. While each of the features assumes knowledge of the range to determine approximate target size, these features are not highly range sensitive. The algorithm calculates each of these features at coarsely sampled ranges between the minimum and maximum allowed range. The features are described below.

Each of the features was chosen based on intuition, with the criteria that they be monotonic and computationally simple. The features are described in decreasing order of importance.

2.2.1 Maximum Grey Level–Feature 0

The maximum grey level is the highest grey level within a roughly target-sized rectangle centered on the pixel. It was chosen because in many FLIR images of vehicles, there are a few pixels that are significantly hotter than the rest of the target or the background. These pixels are usually on the engine, the exhaust manifold, or the exhaust pipe. The feature is defined as

$$F^0_{i,j} = max_{(k,l)\in N_{in}(i,j)} f(k,l), \tag{1}$$

where $f(k,l)$ is the grey level value of the pixel in the kth row and lth column, $N_{in}(i,j)$ is the neighborhood of the pixel (i,j), defined as a rectangle whose width is the length of the longest vehicle in the target set and whose height is the height of the tallest vehicle in the target set. For the applications that we have considered, the width is 7 m and the height is 3 m.

2.2.2 Contrastbox–Feature 1

The contrastbox feature measures the average grey level over a target-sized region and compares it to the grey level of the local background. It was chosen because many pixels that are not on the engine or on other particularly hot portions of the target are still somewhat warmer than the

natural background. This feature has been used by a large number of authors. The feature is defined as

$$F_{i,j}^{1} = \frac{1}{n_{in}} \sum_{(k,l)\in N_{in}(i,j)} f(k,l) - \frac{1}{n_{out}} \sum_{(k,l)\in N_{out}(i,j)} f(k,l), \tag{2}$$

where n_{out} is the number of pixels in $N_{out}(i,j)$, n_{in} is the number of pixels in $N_{in}(i,j)$, $N_{in}(i,j)$ is the target-sized neighborhood defined above, and the neighborhood $N_{out}(i,j)$ contains all of the pixels in a larger rectangle around (i,j), except those pixels that are in $N_{in}(i,j)$.

2.2.3 Average Gradient Strength–Feature 2

The gradient strength feature was chosen because manmade objects tend to show sharper internal detail than natural objects, even when the average intensity is similar. To prevent large regions of background that show higher than normal variation from showing a high value for this feature, the average gradient strength of the local background is subtracted from the average gradient strength of the target-sized region. The feature is calculated as

$$F_{i,j}^{2} = \frac{1}{n_{in}} \sum_{(k,l)\in N_{in}(i,j)} G_{in}(i,j) - \frac{1}{n_{out}} \sum_{(k,l)\in N_{out}(i,j)} G_{out}(i,j), \tag{3}$$

where

$$G_{in}(i,j) = G_{in}^{h}(i,j) + G_{in}^{v}(i,j), \tag{4}$$

$$G_{in}^{h}(i,j) = \left| f(i,j) - f(i,j+1) \right|, \tag{5}$$

$$G_{in}^{v} = \left| f(i,j) - f(i+1,j) \right|, \tag{6}$$

and $G_{out}(i,j)$ is defined similarly.

2.2.4 Local Variation–Feature 3

The local variation feature was chosen because manmade objects often show greater variation in temperature than natural objects. This feature merely determines the average absolute difference between each pixel and the mean of the internal region and compares it to the same measurement for a local background region. The feature is calculated as

$$F_{i,j}^{3} = \frac{L_{out}(i,j)}{n_{in}} - \frac{L_{out}(i,j)}{n_{out}}, \tag{7}$$

where

$$L_{in}(i,j) = \sum_{(k,l)\in N_{in}(i,j)} \left| f(k,l) - \mu_{in}(i,j) \right|, \tag{8}$$

and

$$\mu_{in}(i,j) = \frac{1}{n_{in}} \sum_{(k,l)\in N_{in}(i,j)} f(k,l), \tag{9}$$

and L_{out} (i,j) and $\mu_{in}(i,j)$ are defined similarly.

2.2.5 Straight Edge–Feature 4

The straight edge feature was chosen because manmade object often display straighter temperature gradients than natural objects, especially in the near vertical and horizontal directions. This feature measures the strength of a straight edge that extends for several pixels, and then determines if the edge values in a target sized region differ from the local background. The feature is calculated as

$$F_{i,j}^{4} = \frac{1}{n_{in}} \sum_{(k,l)\in N_{in}(i,j)} H_{in}(i,j) = \frac{1}{n_{out}} \sum_{(k,l)\in N_{out}(i,j)} H_{out}(i,j), \tag{10}$$

where

$$H_{in}(i,j) = H_{in}^{h}(i,j) + H_{in}^{v}(i,j), \tag{11}$$

$$H_{in}^{h}(i,j) = \sum_{|k-i|<l} \left| f(k,j) - f(k,j+1) \right|, \tag{12}$$

$$H_{in}^{v}(i,j) = \sum_{|k-j|<l} \left| f(i,k) - f(i+1,k) \right|, \tag{13}$$

and $G_{out}(i,j)$ is defined similarly. The parameter l is a function of field-of-view of the system, target range, and target size. If the sensor or target is significantly tilted, the functions can be suitably modified to measure edges in other directions, at the cost of more computation and less discrimination ability.

2.2.6 Rectangular Gradient Strength–Feature 5

This feature seeks to take advantage of nearly rectangular target-sized shapes by combining the straight edge strengths that would make up the outer boundary of a roughly rectangular target. The straight edge strengths are defined as above.

The four values to be combined are $H_{in}^{h}(i+l_v.j)$, $H_{in}^{h}(i-l_v,j)$, $H_{in}^{v}(i,j+l_h)$, $H_{in}^{v}(i,j-l_h)$ which correspond to the horizontal top and bottom edges of the target, and the right and left vertical edges. The values l_v and l_h correspond to the half height and half width of the target, respectively. The values are combined using the $L_{1/2}$ norm.

2.2.7 Vertical Gradient Strength–Feature 6

This feature takes advantage of the relative rarity of straight vertical edges in images taken at nearly horizontal viewing angles. The sides of targets are often the most prevelant vertical edges,

though tree trunks and other features sometimes compete. The straight vertical edge image is calculated as before,

$$H_{in}^{v}(i,j) = \sum_{|k-i|<l} \left| f(k,j) - f(l,j+1) \right|, \tag{14}$$

and the local maximum of $H_{in}^{v}(i,j)$over a target-sized region is calculated.

2.2.8 How the Features Were Selected

A full description of the feature selection is outside the scope of this report. A large number of features were programmed, and the value of these features were calculated over a large number of randomly selected pixels in the images of the training set. The feature values were also calculated at the ground truth location of the targets. Histograms were computed for each of the features for both the target and background pixels, and a measure of separability was calculated. The correlation of the features was also calculated to avoid choosing several features that are similar. Some of the features were highly correlated, which was expected because one of the purposes of the training was to determine which of the similar features provided the greatest separability. For example, a number of contrast features were used, which normalized the target and background values by local standard deviation of the background, or of the target, or neither. Similarly, a number of gradient strength features were calculated. The feature pruning process was ad hoc, so it would be reasonable to expect that performance improvement could be obtained by a more rigourous approach.

2.3 Combining the Features

Each feature is normalized across the image so that the feature value at each pixel represents the number of standard deviations from the mean of that feature. Thus the normalized feature image for the *m*th feature is normalized as

$$F_{i,j}^{m,N} = \frac{F_{i,j}^{m} - \mu_m}{\sigma_m}, \tag{15}$$

where

$$\mu_m = \frac{1}{M} \sum_{all(k,l)} F_{k,l}^{m}, \tag{16}$$

and

$$\mu_m = \frac{1}{M} \sum_{all(k,l)} \left(F_{k,l}^{m} - \mu_m \right)^2. \tag{17}$$

After normalization, the features, each of which is calculated for each pixel, are linearly combined into a confidence image,

$$G_{i,j} = \sum_{m=0}^{3} \omega_m F_{i,j}^{m,N}, \tag{18}$$

where the feature weights σ_m are determined using an algorithm not described here. The confidence value of each pixel is mapped by a scaling function $S:\Re \rightarrow [0,1]$, as

$$S\left(G_{ij}\right) = 1 - e^{\alpha G_{i,j}} \tag{19}$$

where α is a constant.

This scaling does not change the relative value of the various pixels, it merely scales them to the interval [0,1] for convenience. Confidence numbers are often limited to this interval because they are estimates of the a posteriori probability. While this is not true for our algorithm, using this interval is convenient for evaluators.

To determine the detection locations from the scaled confidence image, the pixel value with the maximum confidence value is chosen. Then a target-sized neighborhood around the image is set to zero so that the search for subsequent detections will not choose a pixel location corresponding to the same target. The process is then repeated for the fixed number of detections chosen before the algorithm was run.

2.4 Experimental Results

The training results on the Hunter-Liggett April 1992 Region of Interest (ROI) database are shown in the Required Operational Capability (ROC) curve in Figure 1. Test results on the July 1992 ROI database collected at Yuma Proving Grounds is shown in Figure 2, and for Greyling August 1992 ROI database in Figure 3. The Yuma test images are much more difficult, because they were taken in the desert in July, so many locations in the image have a higher apparent temperature than the targets. The images from Greyling, MI are significantly easier because the temperatures are more mild, and are comparable in difficulty to the training data. Note that no training images were used from anywhere but Hunter-Liggett, so the results suggest that the algorithm is not sensitive to the training background. This is not surprising given the simplicity of the algorithm, but sensitivity is common to many learning algorithms. Figures 4 and 5 show a sample image and the results of the algorithm on the image. The cross denotes the ground truth targets, and the x*s* denote the detections on the targets. Detections are designated hits if the detection center falls anywhere on the actual target. Otherwise, they are designated false alarms. The top three detections, ranked by confidence number, are designated on the image. The top two detections are hits, while the third falls near the target and is designated a false alarm. Figures 6 and 7 show another, somewhat more difficult, image and associated algorithm results. The top detection falls on a target in the bottom left of the image, while the second highest detection is a false alarm near the center of the image. The location looks like a possible target; it is merely a warm spot on the dirt road.

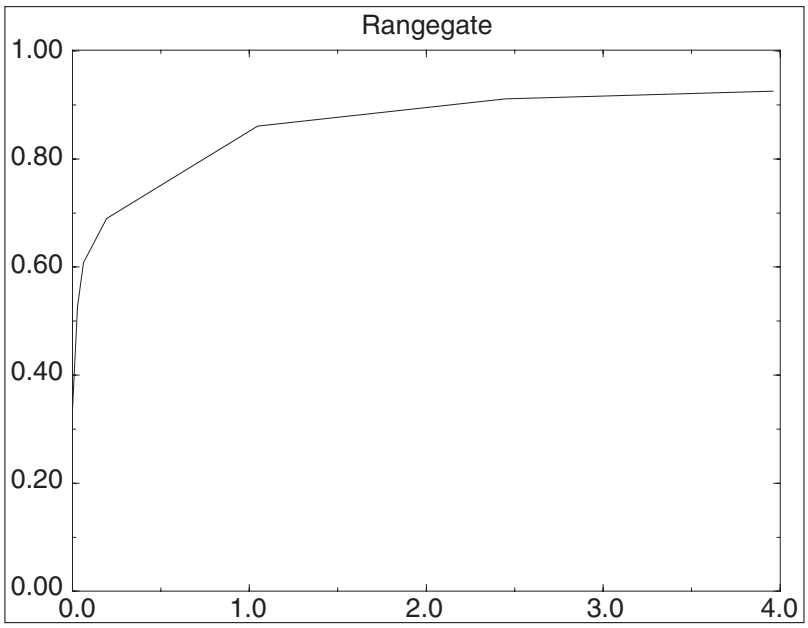

Figure 1. ROC curve on Hunter-Liggett April 1992 imagery. The horizontal axis gives the average number of false alarms per frame, the vertical axis is the target detection probability.

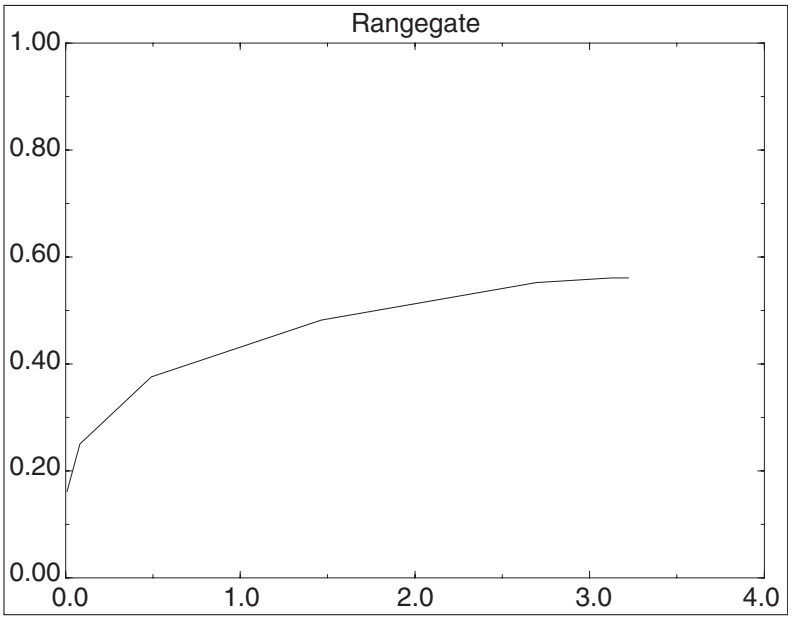

Figure 2. ROC curve on Yuma July 1992 imagery.

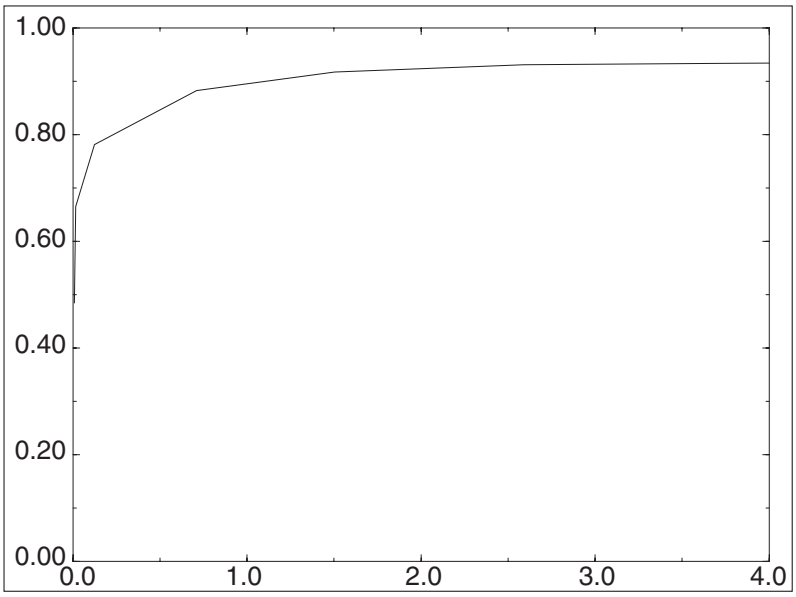

Figure 3. ROC curve on Greyling August 1992 imagery.

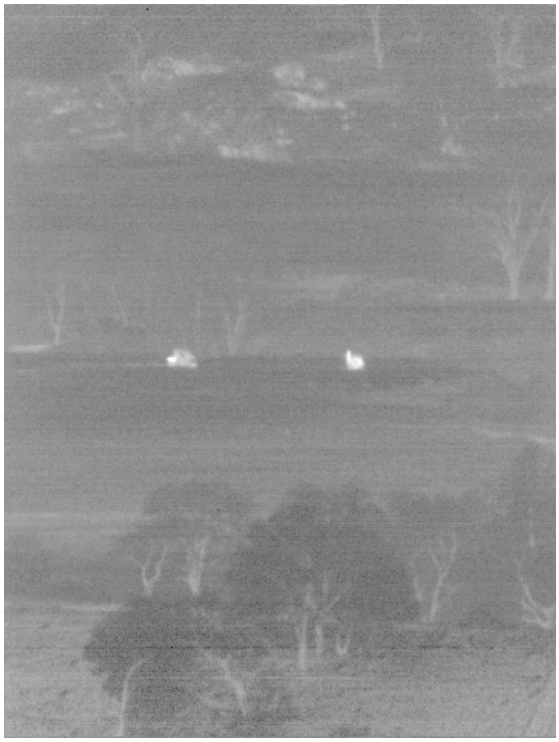

Figure 4. Easy image from Hunter-Liggett April 1992 data set.

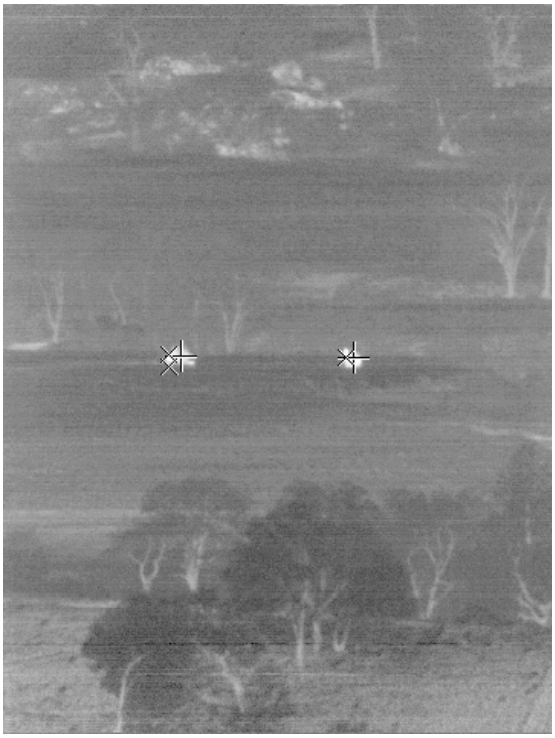

Figure 5. Results on previous image.

Figure 6. Moderate image from Hunter-Liggett
April 1992 data set.

Figure 7. Results on previous image.

The algorithm was also tested on data collected specifically for the DEMO III program to ensure that performance does not degrade because of the different sensor. The DEMO III sensor is sensitive in the midwave, 3-5 μ region, while the previous data was in the longwave, 8-12 μ region. Figure 8 shows an ROC curve on data collected by the DEMO III sensor at the Fort Indiantown Gap DEMO III site.

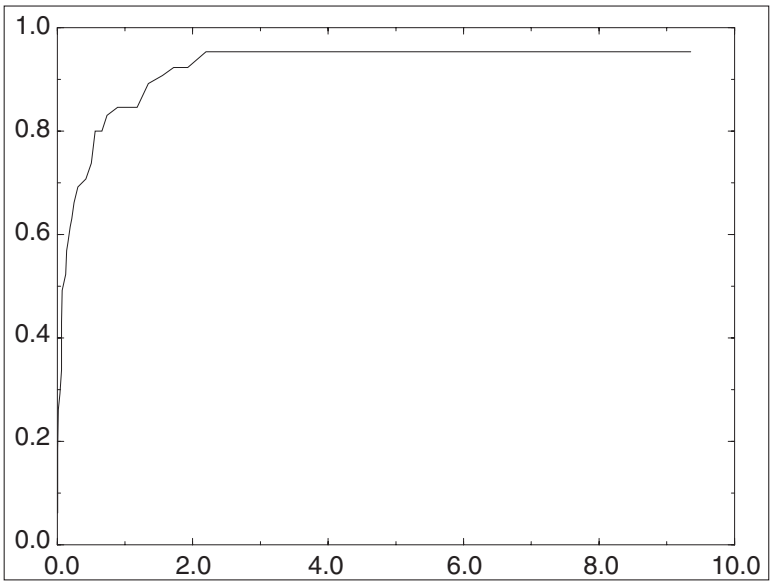

Figure 8. ROC curve on 12-bit 2001 Fort Indiantown gap data.

To determine if raw grey level information could be used to locate targets without the use of shape information, histograms of the 37 Fort Indiantown Gap images were formed. Figure 9 shows the histogram for images that contain no targets, and Figure 10 shows the histogram magnified 100× to show the tail of the distribution. The idea is to determine if the tail for images with targets is higher than for images without targets. Figures 11 and 12 show the corresponding histograms for images with targets. It appears that the raw grey level information would be a poor discriminant for target detection.

The algorithm is being used by the DEMO III program to reduce the amount of imagery that must be trasmitted via radiolink to a human user. It will also be used by the Sensors for UGV program at Night Vision and Electronic Sensors Directorate (NVESD), to prescreen uncooled FLIR imagery and indicate potential targets that should be looked at more closely with an active laser sensor. It has been used by a synthetic image validation tool, by measuring the performance of the algorithm in comparison to real imagery.

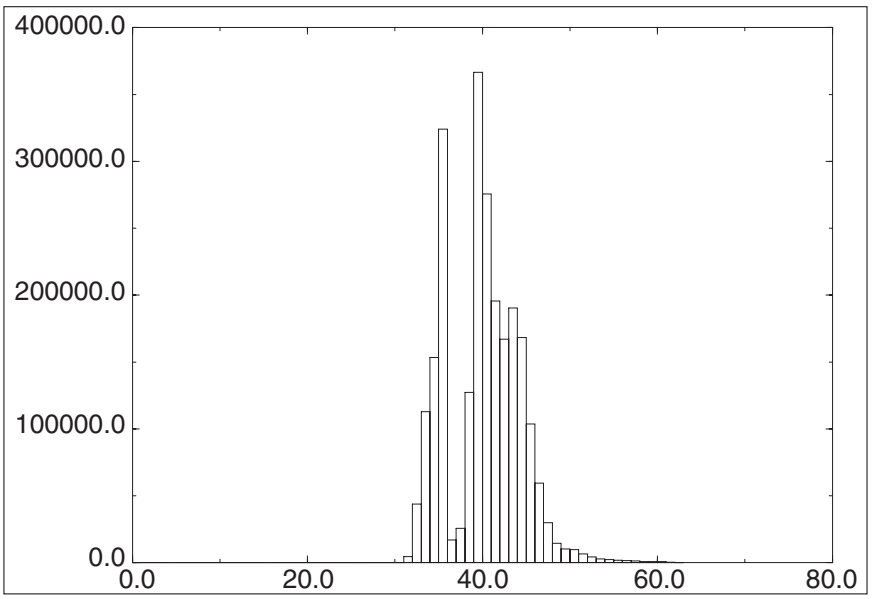

Figure 9. Histogram of grey levels of 37 Fort Indiantown gap images with no targets.

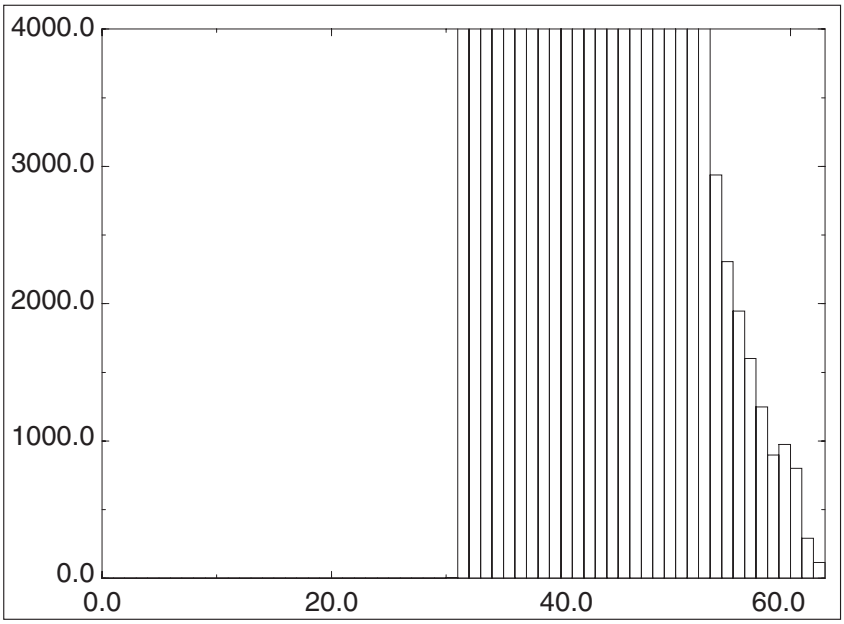

Figure 10. Histogram of grey levels of Fort Indiantown gap images with no targets. The y axis has been magnified 100 times to show tail of distribution.

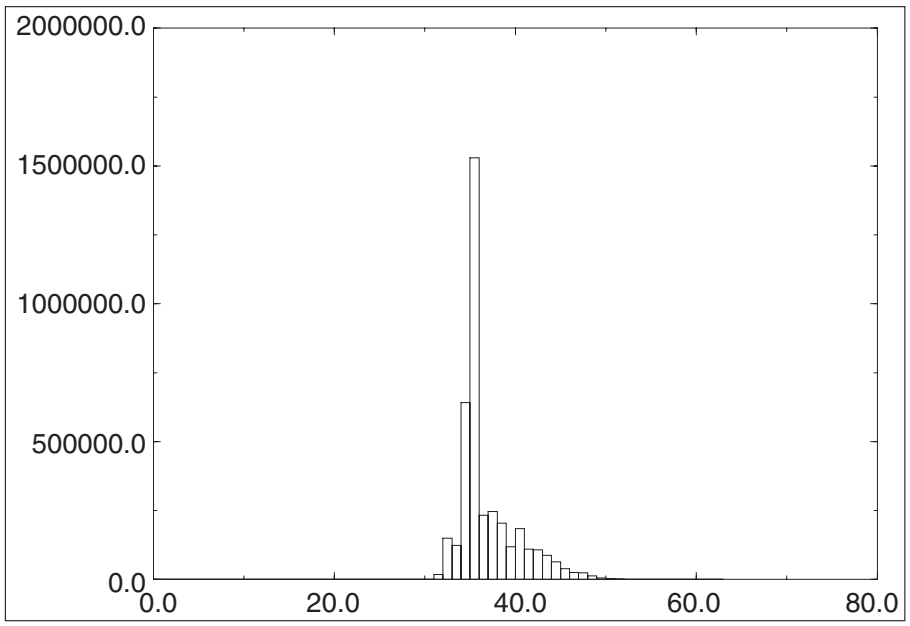

Figure 11. Histogram of grey levels of 60 Fort Indiantown gap images with targets.

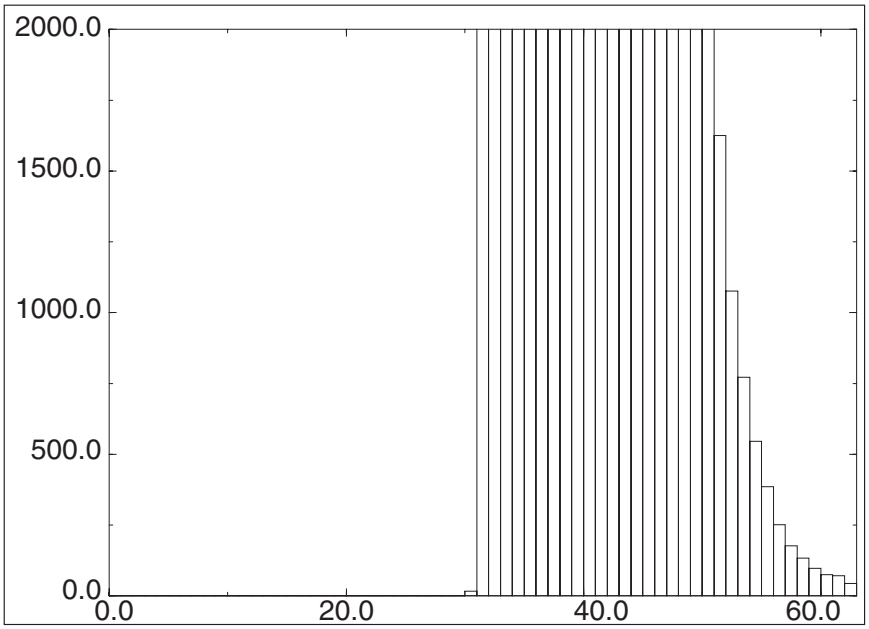

Figure 12. Histogram of grey levels of Fort Indiantown gap images with targets. The y axis has been magnified 100× to show tail of distribution.

3. The Clutter Rejection Algorithm

The purpose of the clutter rejection algorithm is to further examine locations indicated by the detector to determine if targets are present. Because the clutter rejecter doesn't examine the whole image, the algorithm can be more computationally intensive. The algorithm used here is based on PCA-based dimensionality reduction, followed by a multilayer perceptron (MLP) trained to reject clutter and accept targets.

The limited diversity of the training set required that dimensionality reduction be performed before a neural network is used. This is important because using a learning algorithm prior to dimensionality reduction requires a large and diverse training set to avoid overtraining, resulting in a sharp difference between training and testing performance. The architecture of the algorithm has a front-end PCA dimensionality reduction component, followed by a multilayer perceptron that uses only the individual PCA components as inputs. The output of the MLP, along with the feature values from the detector, are combined by a higher level MLP. The following sections describe the PCA and MLP components and describe experimental results.

3.1 PCA

Also referred to as the Hotelling transform or the discrete Karhunen-Loève transform, PCA is based on statistical properties of vector representations. PCA is an important tool for image processing because it has several useful properties, such as decorrelation of data and compaction of information (energy). We provide here a summary of the basic theory of PCA.

Assume a population of random vectors of the form

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_3 \end{bmatrix}. \tag{20}$$

The mean vector and the covariance matrix of the vector population **x** are defined as

$$\mathbf{m}_{\mathbf{x}} = E\{x\}, \text{ and} \tag{21}$$

$$\mathbf{C}_{\mathbf{x}} = E\left\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T\right\}, \tag{22}$$

where E_{arg} is the expected value of the argument, and T indicates vector transposition. Because **x** is n-dimensional, $\mathbf{C}_{\mathbf{x}}$ is a matrix of order $n \times n$. Element c_{ii} of $\mathbf{C}_{\mathbf{x}}$ is the variance of x_i (the ith component of the **x** vectors in the population), and element c_{ij} of $\mathbf{C}_{\mathbf{x}}$ is the covariance between elements x_i x_j of these vectors. The matrix $\mathbf{C}_{\mathbf{x}}$ is real and symmetric. If elements x_{i} and x_{j} are uncorrelated, their covariance is zero and, therefore, $c_{ij} = c_{ji} = 0$. For N vector samples from a random population, the mean vector and covariance matrix can be approximated from the samples by

$$\mathbf{m}_\mathbf{x} = \frac{1}{N}\sum_{p=1}^{N}\mathbf{x}_p\ , \text{ and} \tag{23}$$

$$\mathbf{C}_\mathbf{x} = \frac{1}{N}\sum_{p=1}^{N}\left(\mathbf{x}_p\mathbf{x}_p^T - \mathbf{m}_\mathbf{x}\mathbf{m}_\mathbf{x}^T\right) \tag{24}$$

Because $\mathbf{C}_\mathbf{x}$ is real and symmetric, we can always find a set of n orthonormal eigenvectors for this covariance matrix. Figure 13 shows the first 100 (out of the 800 possible in this case) most dominant PCA eigen-targets and eigen-clutters, which were extracted from the target and clutter chips in the training set, respectively. Having the largest eigenvalues, these eigenvectors capture the greatest variance or energy as well as the most meaningful features among the training data.

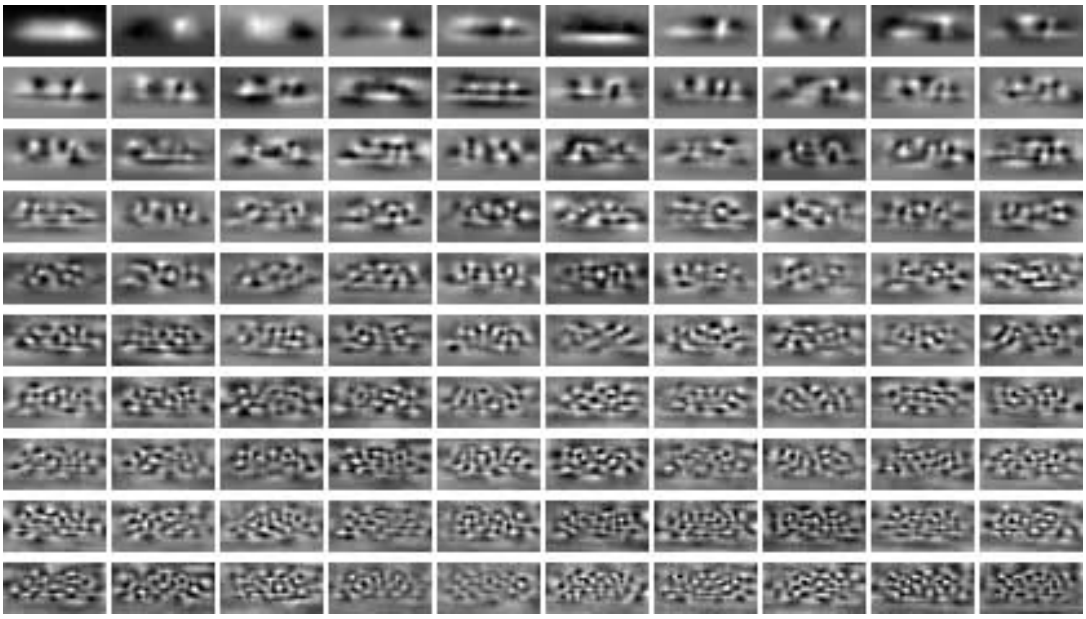

Figure 13. 100 most dominant PCA eigenvectors extracted from the target chips.

Let $\mathbf{e_i}$ and λi, $i = 1, 2, \ldots, n$, be the eigenvectors and the corresponding eigenvalues of $\mathbf{C}_\mathbf{x}$, sorted in a descending order so that $\lambda j \geq \lambda_{j+1}$ for $j = 1, 2, \ldots, n–1$. Let $\mathbf{A}$ be a matrix whose rows are formed from the eigenvectors of $\mathbf{C}_\mathbf{x}$, such that

$$\mathbf{A} = \begin{bmatrix} \mathbf{e_1} \\ \mathbf{e_2} \\ \vdots \\ \mathbf{e_n} \end{bmatrix}. \tag{25}$$

This $\mathbf{A}$ matrix can be used as a linear transformation matrix that maps the $\mathbf{x}s$ into vectors, denoted by $\mathbf{y}s$, as follows:

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})\ . \tag{26}$$

Conversely, we may want to reconstruct vector **x** from vector **y**. Because the rows of **A** are orthonormal vectors, $\mathbf{A}^{-1} = \mathbf{A}^{T}$. Therefore, any vector **x** can be reconstructed from its corresponding **y** by the relation

$$\mathbf{x} = \mathbf{A}^{T}\mathbf{y} + \mathbf{m}_{\mathbf{x}}. \tag{27}$$

Instead of using all the eigenvectors of $\mathbf{C}_{\mathbf{x}}$, we may pick only k eigenvectors corresponding to the k largest eigenvalues and form a new transformation matrix $\mathbf{A}_k$ of order $k \times n$. In this case, the resulting **y** vectors would be k-dimensional, and the reconstruction given in equation (27) would no longer be exact. The reconstructed vector using $\mathbf{A}_k$ is

$$\hat{\mathbf{x}} = \mathbf{A}_k^{T}\mathbf{y} + \mathbf{m}_{\mathbf{x}} \tag{28}$$

The mean square error (MSE) between **x** and $\hat{\mathbf{x}}$ can be computed by the expression

$$\varepsilon = \sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{k} \lambda_j = \sum_{j=k+1}^{n} \lambda_j \,. \tag{29}$$

Because the λ_j's decrease monotonically, equation (29) shows that we can minimize the error by selecting the k eigenvectors associated with the k largest eigenvalues. Thus, the PCA transform is optimal in the sense that it minimizes the MSE between vectors **x** and their approximations $\hat{\mathbf{x}}$. As we can see from figure 13, only the first few score of the eigen-targets contain consistent and structurally significant information pertaining to the training data. These eigentargets exhibit a reduction in information content as their associated eigenvalues rapidly decrease. For the less meaningful eigentargets (say, the 50th and all the way up to the 800th) only high-frequency information is present. In other words, by choosing $k = 50$ in equation (29) when $n = 800$, the resulting distortion error, ε, would be small. While the distortion is negligible, there is a 16-fold reduction in input dimensionality.

3.2 MLP

After projecting an input chip to a chosen set of k eigen-targets, the resulting k projection values are fed to an MLP classifier where they are combined nonlinearly. A typical MLP used in our experiments has $k + 1$ input nodes (with an extra bias input), several layers of hidden nodes, and one output node. In addition to full connections between consecutive layers, there are also shortcut connections directly from one layer to all other layers, which may speed up the learning process. The MLP classifier is trained to perform a two-class problem, with training output values of ±1. Its sole task is to decide whether a given input pattern is a target (indicated by a high-output value of around +1) or clutter (indicated by a low-output value of around –1). The MLP is trained in batch mode using Qprop [7], a modified backpropagation algorithm, for a faster but stable learning course.

Alternatively, the eigenspace transformation can be implemented as an additional linear layer that attaches to the input layer of the simple MLP above. The resulting augmented MLP classifier, which is collectively referred to as PCAMLP network in this paper, consists of a transformation layer and a back-end MLP (BMLP). When the weights connecting the new input nodes to the kth output node of the transformation layer are initialized with the kth PCA eigenvector, the linear summation at the kth transformation output node is equivalent to the kth projection value.

3.3 Experimental Results

The clutter rejection algorithm was trained on data collected at Aberdeen Proving Ground (APG) and Fort Knox in 1999, and tested on data collected at APG in 1999 and Fort Knox in 2000. The APG data were divided so that the training and test data were collected on different days, but the backgrounds are similar because the Perryman test site is rather uniform. ROC curves for the detection/clutter rejection system are shown in Figure 14. The curves show that the clutter rejecter that uses the detector features combined with the output of the MLP by a higher level MLP performs better than the detector alone on the training and test data. Simply using the output of the MLP of the clutter rejector results in worse performance than the detector alone because the MLP does a good job of separating targets and clutter but a poor job of estimating the confidence.

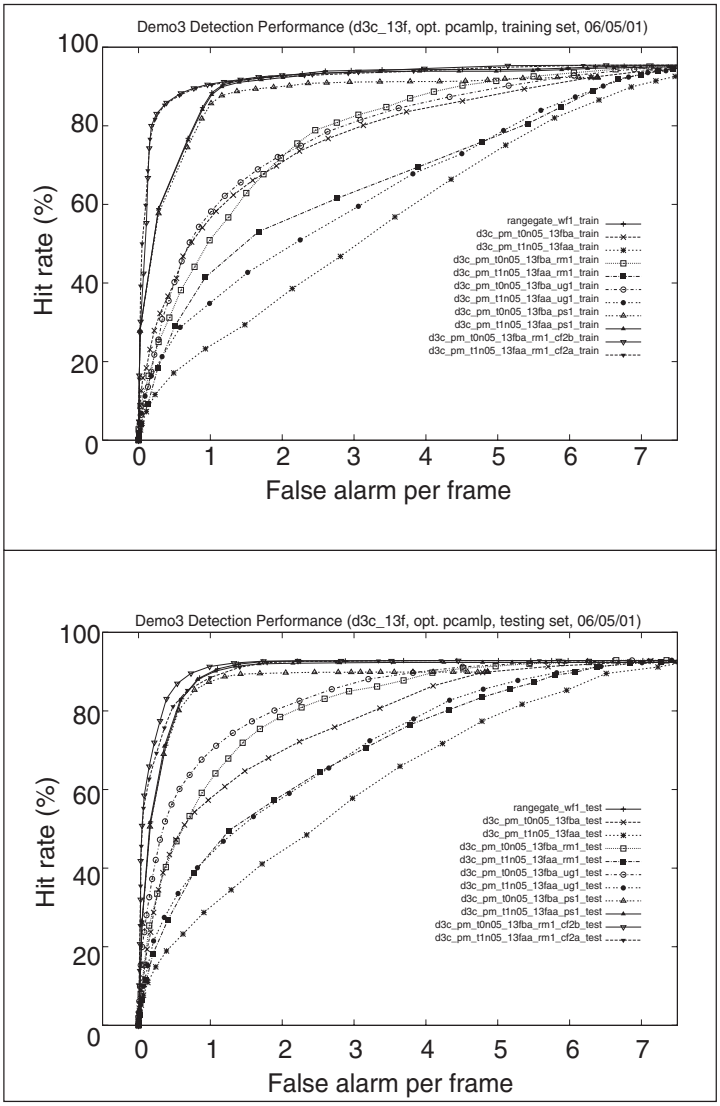

Figure 14. Performance curves.

4. Target Recognition

4.1 Introduction

The algorithm described here was designed to address a need for a recognition algorithm that could be trained with a small amount of data, with poor range and localization information. The operational scenario is to examine objects that have been detected by another algorithm to determine if they are one of the objects stored in an existing image library. The detection location given by the detection algorithm may be poorly centered on the target, and the range to the target will not be known. The number of training examples for the four different targets differed radically. This meant that the chosen algorithm must be able to take advantage of a large training set when it exists but still be able to perform well for smaller data sets.

Many techniques have been applied to the recognition problem [1]. When training sets have been large, recognition algorithms have typically used complex learning algorithms that use a large number of features to discriminate between targets. Often the features are either simply the pixel values, or simple gradient/wavelet features calculated in a dense grid across the target region [3, 4, 8]. The learning algorithms include complex template matching schemes [8] or neural networks [2–4, 9, 10]. Learning algorithms that are trained on small data sets tend to generalize poorly, so we chose not to use these algorithms for this work.

Some algorithm designers have used PCA to compress the data prior to recognition [11]. An advantage of this approach is that it reduces the number of features that a classifier can use, and thus reduces the size of the required training set. One disadvantage is that the compression eliminates some of the information that is useful to perform discrimination, and because the PCA algorithm optimizes the compression of the data without regard for information that is useful for discrimination, one cannot expect that PCA gives the most discrimination information possible for a given number of features.

The data set used for our training was lopsided. The algorithm attempts to recognize four targets, two real (M113 and HMMWV) and two target boards (TB1 and TB2). For the M113 and HMMWV, we have 1239 and 2080 suitable training samples, whereas for the target boards we have 14 and 22 suitable training samples. The ramifications of this imbalance will be discussed.

The remainder of this technote is organized as follows: Section 2 describes the data used to train and test the system. Section 3 describes the architecture of the recognizer. Section 4 gives the results of experiments performed on a small test set of imagery. Section 5 contains conclusions and plans for future work.

4.2 The Data

The training and testing data were gathered from various sources. The testing set consisted of suitable images from the Fort Indiantown Gap data collection of 2001. Images were selected that met a number of conditions. The images must contain the target at sufficient resolution for human recognition. The target images must be nearly unoccluded. Fort Indiantown Gap data were chosen for testing because the actual DEMO scheduled for September 2001 is to be located

there. Also, these data were taken with the same sensor configuration that will be used for the DEMO. The test data contained 64 images of the M113 and 45 images of the target boards.

The training set consisted of images taken with previous configurations of the sensor, or with another sensor. Because the amount of data from the latest sensor configuration was small, we decided to save all of it for testing, and obtain data from other sources for training. The test on the most appropriate data would let us know if the outside data were unsuitable. The only other data of the target boards were obtained with the same sensor, but with an eight-bit digitizer. This resulted in many saturated images; in particular, target regions were likely to be saturated. Other data of the M113 and HMMWV included data from previous versions of the sensor, and from different sensors that we had on hand. For the training data, we had 14 and 22 images of the target boards, and 1239 and 2080 images of the M113 and HMMWV.

The lopsided training set suggests an algorithm architecture that can handle a wide variation of training set size and variability. The numbers given above overstate the problem for a couple of reasons. The target boards are two-dimensional plywood boards with attached heating panels, and as such there is essentially one pose. The real vehicles can be seen from an arbitrary azimuth angle and some variation in elevation. The algorithm groups all of these poses into four groups for the purpose of PCA eigenvector generation. Also, the signature of the target boards is not nearly as variable as the real targets. The target boards have fixed heating panels, so the greatest variability is the angle-of-view variation, and the relative temperatures of the heating panels, the bare plywood, and the background. The solar irradiation on the panel should be nearly constant because the panel is flat. The real target signatures vary because the amount of solar irradiance differs on different portions of the target; the exercise state effects different parts differently (hot wheels if there has been movement, hot engine if engine is running, regardless of movement, etc.), and the pose varies. Still, it can be expected that the training set of the target boards does not capture the variability as well as for the real vehicles, and it is therefore important that a bias reduction technique is used after the PCA transformation.

4.3 Algorithm Architecture

We have chosen a PCA decomposition/reconstruction technique for the algorithm. The idea is to calculate a PCA decomposition of each target-pose group using the training set. For testing, each target is decomposed using the first n PCA eigenvectors, then reconstructed, and the mean square error (MSE) of the difference between the original and reconstructed target is calculated. This gives one MSE value for each target-pose group. The minimum reconstruction error should occur for the correct target-pose group. Because the PCA captures a different proportion of the total information for each of the target-pose groups, the MSE values are adjusted by a weighting vector prior to choosing the minimum value. We emphasize that the data set drives the choice of algorithm architecture.

The PCA decomposition does not capture all of the information in the input target, because the decomposition is truncated at some small number n of eigenvectors. Also referred to as the Hotelling transform or the discrete Karhunen-Loève transform, PCA is based on statistical properties of vector representations. PCA is an important tool for image processing because it has several useful properties, such as decorrelation of data and compaction of information (energy). The basic theory of PCA was described in the clutter rejection section.

4.3.1 PCA Decomposition/Reconstruction Architecture

The PCA decomposition described above takes a training set of images and turns it into an ordered set of eigenvectors and corresponding eigenvalues. This decomposition is performed for each target-pose group of training samples. Because there are two real targets, which are divided into four pose groups each, and two target board types, which only have one pose, there are a total of 10 target-pose groups. We have chosen the number of eigenvectors to retain to be 5, for a total of 50 stored eigenvectors. The eigenvectors are stored at different scales to account for variability in the range to potential targets. The eigenvectors for each of the 10 target-pose groups are shown in Figures 15–24.

Figure 15. Eigenvectors of HMMWV front side.

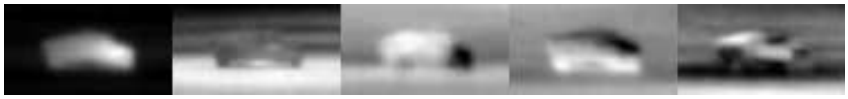

Figure 16. Eigenvectors of HMMWV left side.

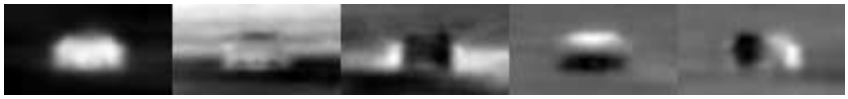

Figure 17. Eigenvectors of HMMWV back side.

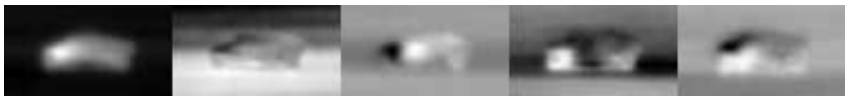

Figure 18. Eigenvectors of HMMWV right side.

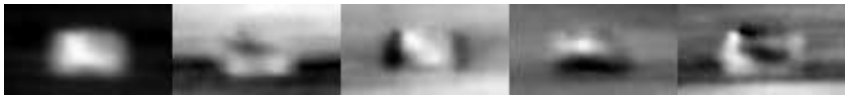

Figure 19. Eigenvectors of M113 front side.

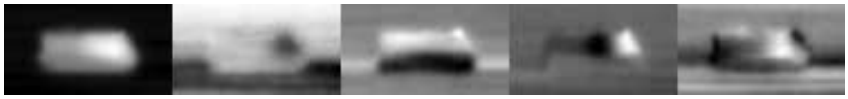

Figure 20. Eigenvectors of M113 left side.

Figure 21. Eigenvectors of M113 back side.

Figure 22. Eigenvectors of M113 right side.

Figure 23. Eigenvectors of target board 1.

Figure 24. Eigenvectors of target board 2.

The decomposition stage determines the PCA components of a sample being tested. The components γ_i for an input target image **x** are calculated as

$$\gamma_i = \sum_{j=1}^{n} \mathbf{e}_{i,j}\mathbf{x}_j = \mathbf{e}_i \cdot \mathbf{x} \,. \tag{30}$$

Thus, the *i*th PCA component of an input vector **x** is simply the dot product of **x** with the *i*th eigenvector. The reconstruction using the first k eigenvectors is

$$\hat{x} = \sum_{i=1}^{k} \gamma_i \mathbf{e}_i \,. \tag{31}$$

The reconstruction error is simply

$$\varepsilon = \sum_{j=1}^{n} \left(\mathbf{x}_j - \hat{\mathbf{x}}_j\right)^2 = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2 \,. \tag{32}$$

4.3.2 Linear Weighting of Reconstruction Error

To reduce the bias inherent in the PCA decomposition process, the reconstruction errors for each target-pose group are multiplied by a fixed weight. Thus, the reconstruction error for the lth target-pose group, ε_l is weighted by a weight ω_l. The target-pose decision $\hat{l}$ is given by

$$\hat{l} = argmin_l(\omega_l \varepsilon_l). \tag{33}$$

The weights ω_l were chosen by experiment.

4.3.3 Scale and Shift Search Space

It is anticipated that this recognizer will be used after an automated detector has found potential targets in an image. It must be assumed that any detector will be imprecise about centering the detection on the target. For the DEMO III application, the range to the target is also unknown, at least for some of the scenarios. Any template matching algorithm is inherently sensitive to translation and scale of the image. The algorithm was written to allow the user to specify the range uncertainty, as well as the translation uncertainty. If accurate range or translation is known, these will help algorithm performance. However, inaccurate information will degrade performance more than lack of information.

The algorithm handles this uncertainty by performing the decomposition/reconstruction operation at a number of different scales and at a few location around the pixel indicated by the detector. Iterating through possible ranges and target locations increases the probability that a false target-pose will give a minimum reconstruction error. The translation uncertainty is specified in pixels. The user is required to specify a minimum and maximum range; if nothing is known about the range to a target, the minimum and maximum ranges can be derived from knowledge of the sensor and knowledge of the minimum resolution required by a recognizer. Range information can be derived from digital maps, shape from motion algorithms, or laser ranging. It is anticipated that for the current implementation, digital maps will be the only regular source of range information.

4.4 Experimental Results

A detection algorithm described elsewhere [12, 13] was applied to the test imagery. The recognition algorithm takes as input the original image and the detection file produced by the detector. The recognizer was not given the ground truth center of the targets, only the detector estimated center. Table 1 shows the confusion matrix on the four class problem. The overall probability of correct identification is 59.63%.

Table 1. Confusion matrix on test set.

	HMMWV	M113	TB1	TB2
M113	6	41	30	14
TB1	1	11	10	1
TB2	0	6	2	14

Figure 25 shows a sample image that does not contain a target. Figures 26–29 contain targets. Some of the targets would be difficult for a human to distinguish. The target in Figure 26 is difficult to distinguish because the shape is not clearly that of an M113, and there is little interior information because the whole target is hot. Figure 27 is clearly an M113 because the rectangular plate on the upper front of the target is a distinguishing characteristic. Notice the target is not level; this makes recognition more difficult because the templates aren't well aligned. The algorithm doesn't currently tilt the templates to handle such a case. Doing so would make it more likely to correctly identify tilted targets but would increase the probability of error on level targets, and would increase computation time. Figure 28 is a good example of a target that is difficult for an algorithm to detect but easy for a human. The target does not have a clear boundary, nor is it hotter than its background. The detector and recognizer give correct results for this image, but that is unusual. Figure 29 shows a target board type II clearly visible in the left center of the image. While this is clearly a target board, it is not easy to see which type at this resolution.

Figure 25. A simple image containing only clutter.

Figure 26. An image of the left side of an M113.

Figure 27. Side view of an M113.

Figure 28. Front view of an M113, on the road near the center of the image.

Figure 29. View of target board type II.

5. Conclusions and Future Work

There is a great deal of work that can still be done to improve the system described. Future work on the detector might include a more systematic evaluation of potential features and an improved classification scheme that allows useful features that appear to be rarely incorporated. In a small

minority of FLIR images of targets, a windshield will reflect cold sky, causing a few pixels to be extremely dark. The current scheme is not set up to incorporate such features because the weighting would be quite low since the feature is seldom useful. The recognizer would greatly benefit from a balanced training set, which would allow for a more sophisticated bias reduction scheme and would enable the formation of a better PCA representation of each target. The algorithms could benefit from more input information. All of the components would benefit from more accurate range information, which could be obtained using accurate registration to digital maps, from structure and motion algorithms, or from a laser range finder. The UGV has a color TV camera collocated with the FLIR, which could provide additional target screening capability during the day.

References

1. Bhanu, B. "Automatic Target Recognition: State of the Art Survey." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 4, pp. 364–379, 1986.
2. Roth, M. W. "Survey of Neural Network Technology for Automatic Target Recognition." *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 28–43, 1990.
3. Hecht-Nielsen, R., and Y.-T. Zhou. "VARTAC: A Foveal Active Vision ATR System." *IEEE Transactions on Neural Networks*, vol. 8, no. 7, pp. 1309–1321, 1995.
4. Wang, L., S. Der, and N. Nasrabadi. "Modular Neural Network Recognition of Targets in FLIR Imagery." *IEEE Transactions on Image Processing*, vol. 7, no. 8, August 1998.
5. Bhanu, B., and T. Jones. "Image Understanding Research for Automatic Target Recognition." *Aerospace Electronic Systems Magazine*, pp. 15–22, October 1993.
6. Jolliffee, I. T. *Principal Component Analysis*, New York: Springer-Verlag, 1986.
7. Fahlman, S. "Faster Learning Variations on Back-Propagation: An Empirical Study." *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, pp. 38–51.
8. Chan, A., and N. Nasrabadi. "Wavelet Based Vector Quantization for Automatic Target Recognition." *International Journal on Artificial Intelligence Tools*, vol. 6, no. 2, pp. 165–178, 1997.
9. Neubauer, C. "Evaluation of Convolutional Neural Networks for Visual Recognition." *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 685–696, 1998.
10. Chen, C. H., and G. G. Lee. "Multi-Resolution Wavelet Analysis Based Feature Extraction for Neural Network Classification." *Proceedings International Conference Neural Networks 3*, pp. 1416–1421, 1996.

11. Chan L., S. Der, and N. Nasrabadi, “Analysis of Dualband FLIR Imagery for Automatic Target Detection.” *Smart Imaging Systems*, Bahram Javidi, (ed.), SPIE Press, 2001.

12. Der, S., C. Dwan, A. Chan, H. Kwon, and N. Nasrabadi. “Scale-Insensitive Detection Algorithm for FLIR Imagery.” ARL-TN-175, U.S. Army Research Laboratory, Adelphi, MD, February 2000.

13. Kwon, H., S. Der, and N. Nasrabadi. “Multisensor Target Detection Using Adaptive Feature-Based Fusion.” *SPIE Aerosense*, April 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 2002	Final, 1999 to 2001

4. TITLE AND SUBTITLE
Automatic Target Acquisition for the DEMO III Program

5. FUNDING NUMBERS
DA PR: AH16
PE: 62120A

6. AUTHOR(S)
Sandor Der, Alex Chan, Gary Stolovy, Michael Lander, and Matthew Thielke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL- SE-SE
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2683

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 2NE4M1
AMS code: 622120.H1600

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes an algorithm for the detection of military vehicles in Forward-Looking Infrared imagery, intended to be used as a prescreener to eliminate large areas of the image from further analysis. The output is a list of likely target locations with confidence numbers that would be sent to a more complex clutter rejection algorithm for analysis. The algorithm uses simple features and is intended to be applicable to a wide variety of target-sensor geometries, sensor configurations, and applications.

14. SUBJECT TERMS
FLLIR, ATR, LA

15. NUMBER OF PAGES
34

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102