ARMY RESEARCH LABORATORY

Automatic Target Recognition Using Wavelet-Based Vector Quantization

by Lipchen Alex Chan and Nasser M. Nasrabadi

ARL-TR-1503

December 1997

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-1503 **December 1997**

Automatic Target Recognition Using Wavelet-Based Vector Quantization

Lipchen Alex Chan and Nasser M. Nasrabadi
Sensors and Electron Devices Directorate

Abstract

An automatic target recognition classifier is described that uses a set of dedicated vector quantizers (VQs) in the wavelet domain. The background pixels in each input image are properly clipped out by a set of aspect windows. The extracted target area for each aspect window is then enlarged to a fixed size, after which a wavelet decomposition is used to split this region into several subbands. A dedicated VQ codebook is then generated for each subband of a particular target class at a specific range of aspects. Thus, each codebook consists of a set of feature templates that are iteratively adapted to represent a particular subband of a given target class at a specific range of aspects. These templates are then further trained by a modified learning vector quantization (LVQ) algorithm that enhances their discriminatory characteristics. Finally, a path selector was designed to speed up the recognition process at the expense of a tolerable degradation in the recognition rate.

Contents

Figures

Tables

1. Introduction

1.1 Background

Human beings are usually very good at recognizing different targets, even in a relatively crowded and changing environment. However, human performance deteriorates drastically in a low-visibility environment or after an extended period of surveillance. Furthermore, certain working environments are either inaccessible or too hazardous for human beings. To compensate for such limitations of human operators, an accurate and versatile automatic target recognition (ATR) system is needed. For example, an ATR system in a battlefield could alert graveyard-shift watchmen with accurate information about any approaching vehicle, so that appropriate responses could be made in a timely fashion. Similarly, a robust ATR system could reduce the workloads of fighter pilots or tank commanders significantly by suggesting effective responses in real time. In the civilian sector, mission-specific ATR systems have been constructed for a number of tasks, including autonomous vehicle navigation, automobile manufacturing and inspection, and orchard sprayer systems in agriculture [1].

Despite their diversity, all ATR applications require an efficient and reliable target recognition method. Unfortunately, the development of such a method is often hampered by the large number of target classes and aspects, long viewing ranges, obscuration, high-clutter background, different geographic and weather conditions, sensor noise, and variations caused by translation, rotation, and scaling of the targets. Furthermore, a range of factors—similarities between the signatures of different targets, limited training and testing data, camouflaged targets, nonrepeatability of target signatures, and the difficulty in using the contextual information whenever it is available to the recognition system—make the recognition problem even more challenging. ATR applications for military purposes are especially susceptible to these challenges, because in comparison to civilian applications, military applications tend to be operated at a wider range of hostile conditions, and it is much harder to collect an adequate and suitable set of data for training and testing purposes. Similar difficulties also occur in other recognition tasks, such as human face recognition [2,3] and handwriting recognition [4,5].

In this report, we use Comanche second-generation forward-looking infrared (FLIR) image chips as our training and testing sets. These images were collected at different sites (Ft. Hunter-Liggett, CA; Yuma, AZ; and Ft. Grayling, MI), seasons (winter and summer), times of day (day and night), and operational conditions of the target (hot and cold). These data are assumed to have come from an "unfamiliar environment" (according to the definition given by Dasarathy [6]), because the identification of the training data might not be reliable (with a level of reliability that is not known a priori). Owing to the inherent characteristics of a FLIR sensor, the signatures of the targets within the scene are severely affected by rain, fog, and foliage [7]. Fortunately, a number of FLIR image enhancement techniques can be used to preprocess the FLIR input images before detection and recognition. Lo [8] examined six of these techniques: threshold zonal filtering, statistical differencing, unsharp masking, prototype automatic target screener technique, constant variance, and histogram equalization. He found that the variable threshold zonal filtering technique performed most satisfactorily, followed by the prototype automatic target screener technique and unsharp masking.

A complete ATR system may consist of several algorithmic components, such as preprocessing, detection, segmentation, feature extraction, classification, prioritization, tracking, and aim-point selection [9]. In this report, we assume that the locations of the potential targets are determined a priori by a high-performance target detection algorithm (a cuer) with a very low false-alarm rate. An example of such a detection algorithm is the ATR relational template matching (ARTM) algorithm proposed by Kramer *et al* [10]. The boxes in figure 1 indicate the potential target areas (target chips) that were detected by the ARTM algorithm. Our focus is to correctly classify each target in these locations into one of the known target types, which essentially is a target classification or recognition task. Therefore, the actual inputs to our classifier are target chips that were extracted from the detected regions and assumed to be clutter-free. Examples of fairly good quality target chips for a truck at various viewing aspects are shown in figure 2. For a given input image, the outputs of our classifier are the class likelihoods for all target types considered. These outputs can then be used by later stages, such as prioritization and aim-point selection.

Generally, a target recognition problem can be attempted with one or a combination of the statistical, structural/syntactic, and neural networks approaches [11]. For example, the *nearest neighbors* or *K-means* unsupervised classification methods are typical examples of nonparametric statistical techniques, which assume that one can define K reasonable clusters for a given data set by minimizing a distance measure between the data and

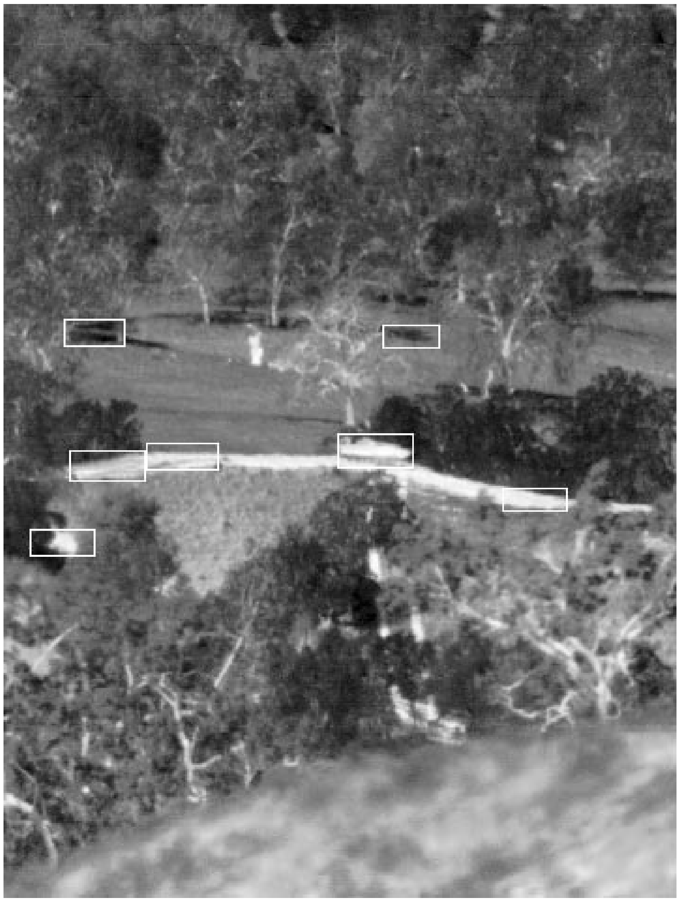

Figure 1. An example FLIR image taken in a typical environment. Boxes indicate potential target areas detected by ARTM algorithm.

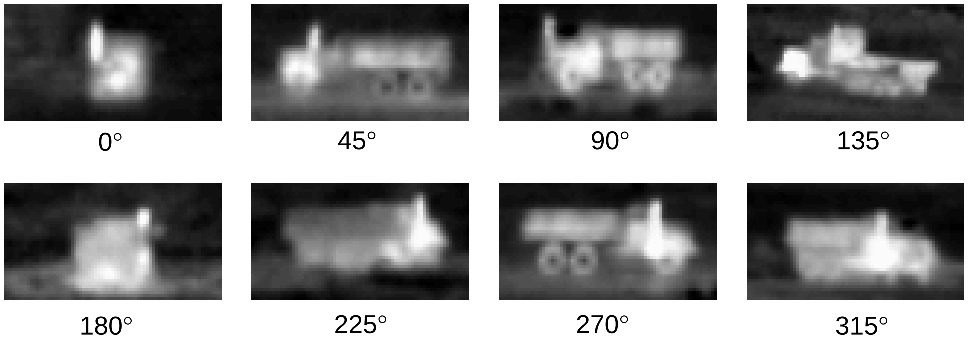

Figure 2. Selected target chips of a truck at various aspects.

the *centroids* of the clusters [12]. A number of studies have been carried out to improve the learning rule and performance of these clustering techniques [13,14]. A supervised sibling of the K-means clustering technique is the learning vector quantization (LVQ) algorithm [15,16]. In the LVQ algorithm, a class identity reference vector is available for each training sample, and the clustering is performed with respect to a distance measure along with the class label. Besides these traditional recognition methods, model-free neural network based methods are also gaining popularity because of their learning capability and massively parallel implementation [17–23].

Recently, we proposed an ATR classifier that employed both the statistical and neural network approaches [24,25]. In that classifier, a number of vector quantizers (VQs) and multilayer perceptrons (MLPs) were modularly cascaded to perform the target recognition task. The inputs to the VQs in that classifier were image blocks extracted from the target chips in the spatial domain. However, because of the high dimensionality of input images and the scarcity of the training data, it is often necessary to further reduce the data dimensionality by transforming the input data into a more compact feature space before the classification process. For example, in a texture classification task, McLean [26] first transformed the input image block into the spatial frequency domain with a discrete cosine transform (DCT). With these local spatial frequency features, he performed the transform vector quantization for the combined purpose of texture coding and classification. Besides the DCT, principal component analysis (PCA) [27] and the most discriminating features (MDF) method [28] are among the other techniques that have been used for dimensionality reduction in a target recognition task. In this report, we reduce dimensionality using a wavelet decomposition process [29].

In many situations, it is quite beneficial to break up a complex classification task into several smaller and easier subtasks. For instance, Anand *et al* [30] used a modular network architecture to reduce a K-class classification problem into K two-class problems, with a separately trained network for each two-class problem. With this decomposition of task complexity, they reported a faster convergence on the simpler modules and noted the feasibility of parallel processing. As the computing hardware and software have become more available for parallel processing, many researchers have proposed massively parallel computing architectures that are specifically optimized for image processing [31–33]. We incorporate modularity into our classifier by constructing functionally similar processing paths and allowing them to operate in parallel and independently from each other.

1.2 Research Objectives

The goal of the proposed ATR classifier described in this report is to recognize military targets in FLIR imagery. The schematic diagram, shown in figure 3, shows the four stages of our classifier: a set of aspect windows of different size, a stage in which the extracted area is enlarged to a fixed size, a stage for wavelet decomposition of the enlarged extraction, and a dedicated VQ for each subband within each aspect window.

In the first stage, an aspect window is a background-clipping rectangle whose size is determined by the type of target and the range of aspects that it operates on. These aspect windows are needed for accurate extraction of the target pixels from the input image, so that the irrelevant background pixels that carry little information about the target are removed before further processing occurs. After the background removal in the first stage, each extracted target area is enlarged in the second stage into a fixed dimension that is common to all the aspect windows. In the third stage, the enlarged extraction is decomposed into four subbands based on a wavelet decomposition process [34]. This decomposition process subdivides the complexity of the recognition task and reduces the dimensionality of the VQ in the following recognition stage. The generalization capability is also improved through

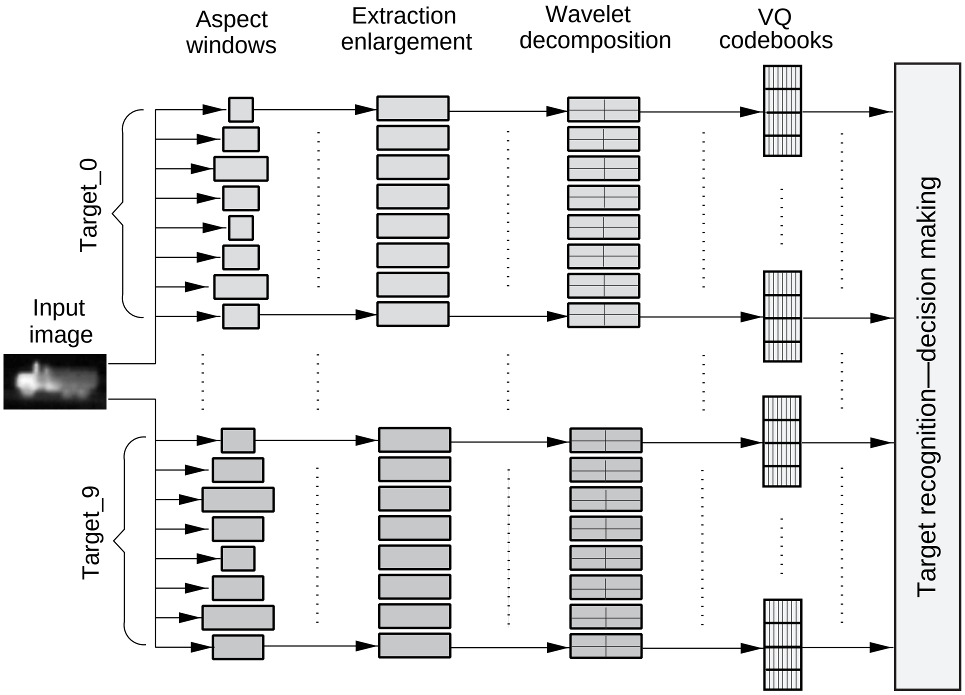

Figure 3. Proposed automatic target recognition classifier.

various manipulations of these orthogonal subbands. After the wavelet decomposition, the final stage uses a set of VQ codebooks for feature matching and target recognition. In this stage, the target recognition problem is treated as a template matching task through a similarity-metric–based approach. A set of subband templates or *code vectors* is constructed for each subband of a particular target at a specific range of aspects. Each set of code vectors forms a *codebook*, representing the target signatures for a given subband of a particular target at a specific range of aspects.

During the testing phase, each subband of the extracted target area is represented by a similarity measure that compares the given subband with the best-matching code vector from the corresponding codebook. A commonly used similarity measure is the mean squared error (MSE). We can infer the class of an input image by the MSEs obtained from comparing all the subbands with the code vectors in the corresponding codebooks. The class of the subband codebooks that produce the smallest overall MSE is expected to be the class of the input image.

In section 2, we discuss in detail each component of the proposed ATR classifier and the algorithms for training them. Experimental results are presented in section 3. A brief comparison of our recognition results with the performance of another compatible ATR classifier is provided in section 4. Finally, some conclusions are given in section 5.

2. Wavelet-Based ATR Classifier

2.1 Aspect Windows and Extraction Enlargement

Normally a target is surrounded by some background information that is irrelevant for the correct recognition of the target. This background information tends to create unwanted variations during the training phase, which could later become problematic noise in the testing or the recognition phase. Therefore, proper removal of the unwanted background is essential for achieving good recognition performance.

The size and shape of target silhouettes often differ significantly at different viewing aspects. For each target, the classifier uses several rectangular windows of different size in order to remove the background information. For example, by using the ground-truth silhouettes that are generated by computer-aided design (CAD), the classifier can cluster the silhouettes of each target into three different window categories representing the front/rear, oblique, and side views of the target; thus, a total of eight aspect windows is needed (one front view, one rear view, two side views, and four oblique views), as shown in figure 4. Because the height of a particular target does not change over the viewing aspects, the window clustering process is based solely on the width of silhouettes. We use the K-means algorithm to perform this clustering process. Since the size of the silhouettes for the side views (around 90° and 270°) changes relatively slowly at different viewing aspects, the classifier uses a broader range of aspects for the side aspect windows (a range of 70°) than for the head and tail aspect windows (a range of 40°). Assuming that the target in each image has been shifted to the center of the image and scaled to a fixed range, the unwanted background is properly clipped away by these aspect windows. An example of this background clipping is illustrated in figure 5. In the proposed algorithm, the segmentation of the target from the background and the target classification are performed simultaneously. This process differs from many existing ATR methods that separately perform target segmentation followed by a classification of the segmented area. Obviously, such classifiers need a correct target segmentation for a successful recognition by the subsequent stages.

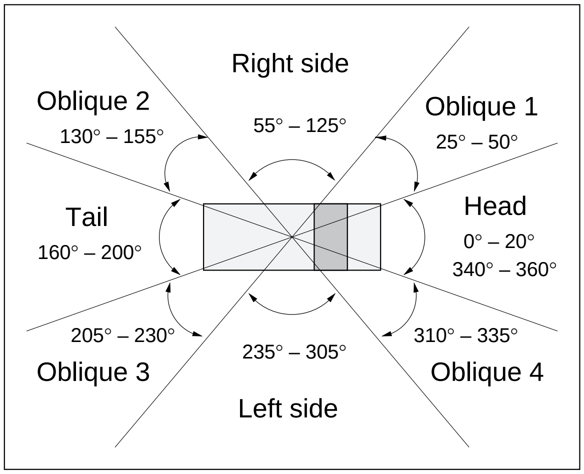

Figure 4. Partition of aspects into eight sets for a target chip.

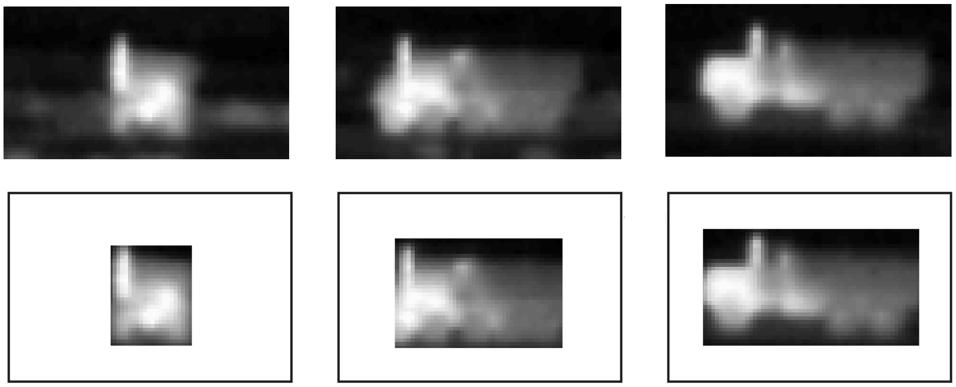

Figure 5. Background clipping of several input images. (Top) Original images of a truck viewed at 0°, 45°, and 90°, respectively. (Bottom) Corresponding extracted target areas after proper background removal.

After the background removal, the extracted target area is enlarged to a fixed size. This fixed dimension should be slightly larger than the largest aspect window, so that interpolation can be used for all the aspect windows, and no crucial information in the extracted area will be lost by the enlargement process. This enlargement enables a more uniform similarity measure in terms of dimensionality, and hence legitimizes the comparison between winning code vectors from different aspect windows during the LVQ training and testing phases. Without this enlargement, smaller aspect windows may have an advantage in finding a good match with an input image, even if that input image indeed belongs to a bigger aspect window.

2.2 Wavelet Decomposition

Wavelets are mathematical functions that separate data into several different frequency components, and then represent each component with a resolution matched to its scale. Compared to traditional Fourier methods, wavelets are more suitable in analyzing physical situations where the signal contains discontinuities and sharp spikes [34]. The biggest difference between these two kinds of transforms is that the individual wavelet functions are localized in space, while the Fourier sine and cosine functions are not. Because of the wavelets' space-localization property, many functions are *sparse* when transformed into the wavelet domain. Because of this sparseness, wavelets have been shown to be very useful in detecting features in images [35,36], image compression [37–39], texture discrimination [40], removal of noise from time series [41], and so forth.

The quadrature mirror filter (QMF) was first introduced by Croisier *et al* [42] as a tool that allows alias-free reconstruction of the signal in the absence of quantization errors. More recently, Vetterli and Herley [43] described the relationships among wavelets, filter banks, and multiresolution signal processing in greater detail. We implemented the wavelet decomposition in this report through QMF using the simplest Haar filter family [44]. Because of the small size of the target chips in our experiments and the efficiency of the ATR classifier, only the Haar two-tap even-length filters were used. In spite of its simplicity and differential discontinuity, the Haar basis can still perform reasonably well in an image compression task if it has good impulse and step response properties, as in the case of two-tap filters (Villasenor *et al* [45]). Villasenor *et al* also found that even-length filters have significantly less shift variance than odd-length filters and possess superior impulse response performance, so that even-length filters can perform much better in preserving the location, shape, and intensity of impulses (sharp edges). These properties make the Haar two-tap filters well suited to a target recognition task.

Based on the wavelet decomposition method described above, the enlarged extracted area of each aspect window is decomposed into four subbands of equal size. An example of the decomposition is shown in figure 6. We normalize each subband by subtracting the mean of that subband and then dividing each pixel by its standard deviation. This way, unwanted variations among similar samples of a particular subband, such as differences in brightness and contrast, can be reduced. This normalization step is critical to securing consistent input information for the VQ.

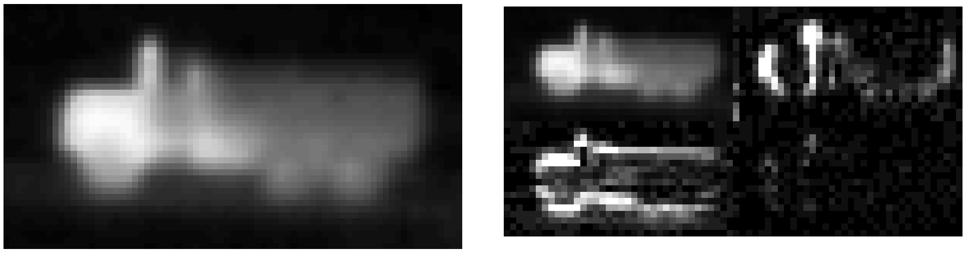

Figure 6. Wavelet decomposition of a truck (left) into four subbands using Haar two-tap filters (right).

2.3 Vector Quantizers

The *Voronoi* or nearest neighbor VQ belongs to a special class of VQ whose partition is solely determined by a codebook and a distortion measure [46]. If the MSE distortion measure $d(\mathbf{X}, \mathbf{Y})$ is defined as the mean squared Euclidean distance between two vectors $\mathbf{X}$ and $\mathbf{Y}$, of dimension k, such that

$$d(\mathbf{X}, \mathbf{Y}) = \frac{1}{k} \, ||\mathbf{X} - \mathbf{Y}||^2 = \frac{1}{k} \sum_{i=1}^{k} (x_i - y_i)^2 \ ,$$

then a partition cell R_i of a Voronoi VQ of L levels is defined as

$$R_i = \{ \mathbf{X} : d(\mathbf{X}, \mathbf{Y}_i) \leq d(\mathbf{X}, \mathbf{Y}_j) \mid j = 1, 2, \ldots, L; \ j \neq i \} \ ,$$

where R_i consists of all the points $\mathbf{X}$ in R^k space that have the least distortion when reproduced with the code vector $\mathbf{Y}_i$. All partition cells are formed by the intersections of half-spaces that are determined by the code vectors explicitly.

For each aspect window, four independent codebooks are constructed, one for each subband. The number of levels L for each subband codebook is determined by the variability of the information within that subband. The *total MSE* for an aspect window is a function (such as a simple summation) of the best MSEs produced by the code vectors from the four codebooks associated with that aspect window. Since the extracted target area is enlarged to a fixed dimension, the code vector size k is the same for all the codebooks in all the aspect windows. Therefore, the total MSE measure can legitimately be used for distortion comparisons between the codebooks of different aspect windows.

The resulting VQ codebooks are indeed a powerful form of constrained VQ, which is usually referred to as a *product code* VQ [46]. To visualize the idea, assume an enlarged extraction $\mathbf{W}$ of dimension $m > 1$. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_b$ be a set of subbands that are functions of $\mathbf{W}$ and jointly determine $\mathbf{W}$.

There are functions f_i for $i = 1, 2, \ldots, b$ such that $\mathbf{W}$ can be decomposed into subband $\mathbf{X}_i$ according to $\mathbf{X}_i = f_i(\mathbf{W})$ for $i = 1, 2, \ldots, b$. Each $\mathbf{X}$ is sometimes called a *feature vector* because it represents some characteristics of $\mathbf{W}$ while still partially describing $\mathbf{W}$. Each feature vector assumes values in a more compact region of m-dimensional space or has a lower dimensionality, and hence should be easier to quantize than $\mathbf{W}$. For each i, let C_i be a codebook with N_i code vectors, which contains the reproduction values for $\mathbf{X}_i$. A *product code* VQ is then a VQ that finds the indices of the closest centroids $\hat{\mathbf{X}}_i \in C_i,\ i = 1, 2, \ldots, b$, in order to completely represent $\mathbf{W}$. The resulting set of indices is called a product code, because the requirement that $\hat{\mathbf{X}}_i \in C_i$ for all i is equivalent to saying that the overall reproduced vector $(\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_b)$ is in the space defined by the Cartesian product C of the b codebooks, that is, $(\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_b) \in C = C_1 \times C_2 \times \cdots \times C_b$. The set of all possible reproduction vectors for $\mathbf{W}$ is then defined by the set of all possible combinations taking any code vector from each of the b codebooks, that is, $N = \prod_{i=1}^{b} N_i$ possible reproduction vectors in all. The $\mathbf{W}$ is indeed encoded with this huge set of reproduction vectors. However, this codebook is not optimal in general, because its code vectors are constrained by the structure of the reproduced overall vector $(\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_b)$ and the individual component codebooks C_i.

For these Voronoi VQs, the training is performed independently for each codebook. Only the training data that belong to a given target class and aspect window are used to train the codebooks that are dedicated to the subbands in that target class and aspect window. To construct and initialize the codebooks, we cluster the training data for a given codebook according to a predefined cluster boundary. Clusters or code vectors with a very low population are discarded, so that the codebook is sufficiently small but still contains the more popular feature templates. Nonetheless, we keep at least 10 code vectors for each codebook in order to maintain a minimal distinguishing capability and to compensate for any overly small cluster boundary.

The K-means algorithm [47] is used to train each Voronoi VQ independently by updating each code vector in each codebook with the average of all the data that are closest, in terms of Euclidean distance, to that code vector. The goal of this learning process is to capture, by minimizing the average distortion, the contextual similarities among the samples that belong to a particular subband of the intended target and aspect window. The training stops when no more changes have been made to any of the code vectors. After the K-means training, the code vectors of a codebook will represent the most general structures extracted for all the input targets that belong to that particular subband of a given target class and aspect window. Figure 7 shows all the code vectors of the four codebooks that belong to the left view of a truck, after the K-means training process. The edges of the target are relatively blurred in these code vectors, because similar input images do not always have the same sharp edges and rarely occur at the same location.

Since the average distortion is minimized, a Voronoi VQ works fairly well in data compression. However, this type of VQ does not perform well as a classifier since different classes often overlap in the feature space. As a result, the independently trained codebooks can have very similar code vectors, which

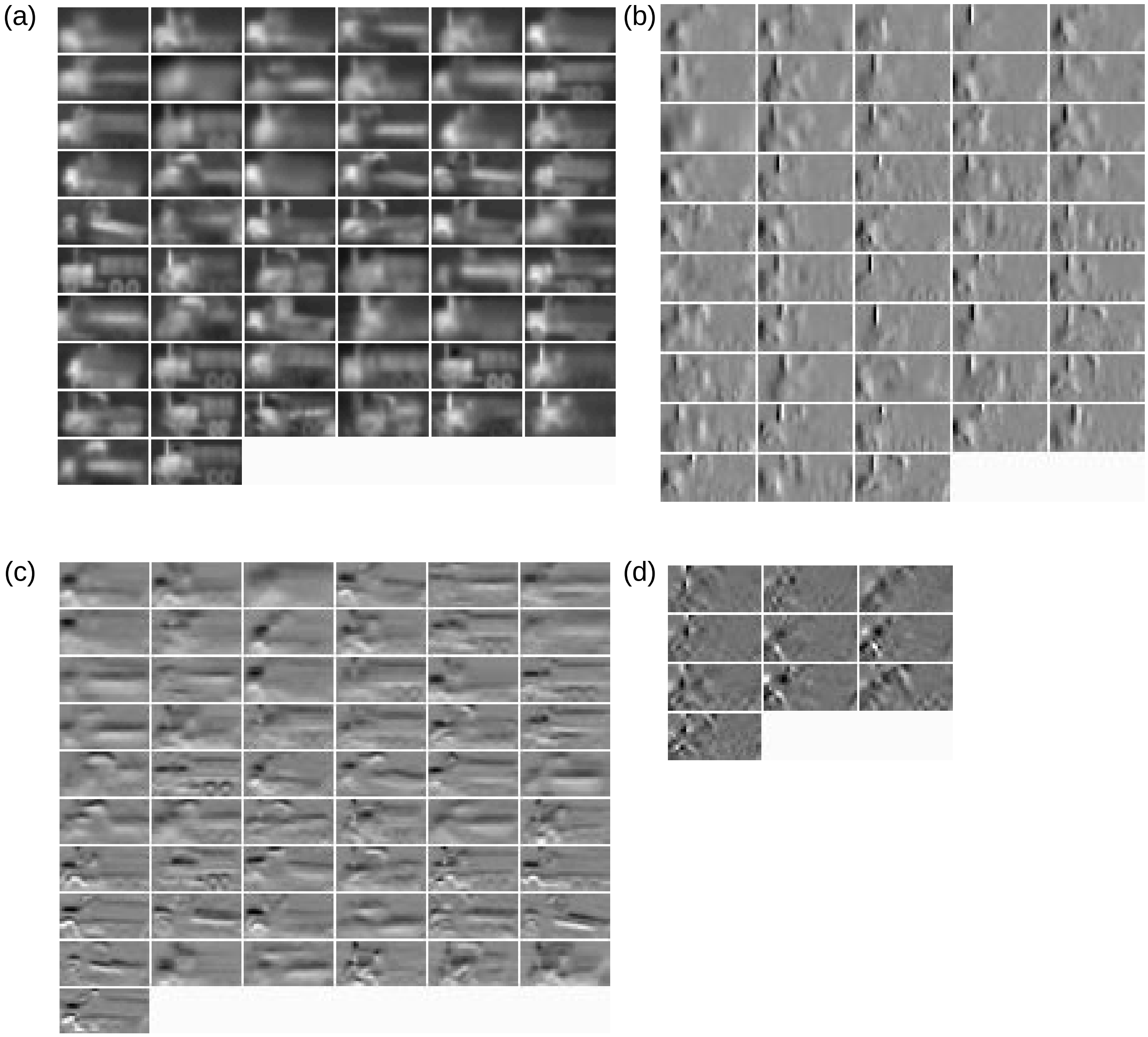

Figure 7. Content of codebooks for left view of a truck after K-means training process: bands (a) LL, (b) HL, (c) LH, and (d) HH. (L refers to low-pass decomposition and H to high-pass decomposition.)

may lead to misclassification when ambiguous input images are encountered. A better method is needed to cleverly alter the half-spaces of those overlapping partitions that originated from different Voronoi codebooks, so that the recognition performance can be improved.

2.4 Learning Vector Quantization

One of the popular methods to globally modify the decision boundaries of a Voronoi VQ is supervised LVQ. Many variants of LVQ have been proposed and applied for classification purposes, as was shown by Kohonen [15,16]. The version of the LVQ algorithm proposed in this report is modified from a variant called LVQ2.1, proposed by Kohonen [15]. In LVQ2.1, the updates are performed on two code vectors, $\mathbf{Y}_i$ and $\mathbf{Y}_j$, that are nearest to a training image block $\mathbf{X}$, provided that one of these code vectors belongs to the correct class, while the other belongs to a wrong class, and $\mathbf{X}$ falls within an *update zone* defined around the mid-plane of $\mathbf{Y}_i$ and $\mathbf{Y}_j$. Assuming that d_i and d_j are the Euclidean distances of $\mathbf{X}$ from $\mathbf{Y}_i$ and $\mathbf{Y}_j$, respectively, this update zone is defined as the region where

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > T ,$$

and T is a threshold whose value is usually chosen between 0.5 and 0.7, depending on the application. This update zone restricts the updates to only those highly ambiguous pairs of code vectors that are around the class decision boundaries. Assuming $\mathbf{X}$ belongs to the same class as $\mathbf{Y}_i$ and not to $\mathbf{Y}_j$, the two code vectors are updated as follows:

$$\begin{aligned}\mathbf{Y}_i(t+1) &= \mathbf{Y}_i(t) + \alpha(t)[\mathbf{X}(t) - \mathbf{Y}_i(t)] ,\\ \mathbf{Y}_j(t+1) &= \mathbf{Y}_j(t) - \alpha(t)[\mathbf{X}(t) - \mathbf{Y}_j(t)] .\end{aligned}$$

This iterative process will decrease the Euclidean distance for $\mathbf{Y}_i$ and increase it for $\mathbf{Y}_j$. The α is the learning rate, usually with an initial value around 0.1, which may be decreased gradually in the course of training.

We now explain the modified LVQ algorithm that is used in our target classification technique. Suppose that we have a classification problem consisting of K target classes and W aspect windows for each class. Let the enlarged extracted target area be decomposed into B subbands for each aspect window, and say that there are P patterns in the training set. Several variables are defined and the modified LVQ algorithm is summarized as follows:

The variables $\{\,\mathbf{X}_{pkwb},\ \mathbf{C}_{kwb},\ L_{kwb} \mid p = 1,2,\ldots,P;\ k = 1,2,\ldots,K;\ w = 1,2,\ldots,W;\ b = 1,2,\ldots,B\,\}$ denote the subband b of the aspect window w

of class k for the training pattern p, the codebook dedicated to the subband b of aspect window w of target class k, and the number of code vectors in $\mathbf{C}_{kwb}$, respectively. Also, the variables $\{\,\mathbf{V}_{kwbl},\ \mathbf{A}_{kwbl},\ U_{kwbl},\ E_{kw} \mid k = 1,2,\ldots,K;\ w = 1,2,\ldots,W;\ b = 1,2,\ldots,B;\ l = 1,2,\ldots,L_{kwb}\,\}$ denote the l^{th} code vector in the codebook $\mathbf{C}_{kwb}$, the total error gradient accumulated for updating $\mathbf{V}_{kwbl}$, the frequency of such accumulations in an epoch, and the total MSE produced by the aspect window w of target class k for a given input pattern.

Step 1: Given the training input pattern p' that belongs to the window w' of class k', compute the best total MSE obtained by matching all the B subbands with the best-matching code vectors in their corresponding codebooks:

$$E_{kw} = \sum_{b=1}^{B} \left(\min_{l=1}^{L_{kwb}} ||\mathbf{X}_{p'kwb} - \mathbf{V}_{kwbl}|| \right) ,$$

for $k = 1,2,\ldots,K;\ w = 1,2,\ldots,W$. Functions other than a simple summation can also be used in the calculation of E_{kw}, such as summing only the best $b < B$ MSEs. For this input pattern, find E^*, the minimum total MSE obtained among all the aspect windows, and E^t, the total MSE produced by the codebooks that belong to the window w' of class k':

$$E^* = \min_{\forall kw} E_{kw} \quad \text{and} \quad E^t = E_{k'w'} .$$

Now compute the *distance for the updating neighborhood*, D, which is a function of E^*, such as

$$D = E^* \times \text{constant} .$$

Step 2: For each E_{kw}, if $E^t < D$ and $E_{kw} < D$ while both E^t and E_{kw} are one of the *four* smallest total MSEs for this pattern, then for $b = 1,2,\ldots,B$, calculate

$$\begin{aligned}
\mathbf{A}_{kwbl'} &= \mathbf{A}_{kwbl'} + \alpha(\mathbf{X}_{p'kwb} - \mathbf{V}_{kwbl'}) && \text{if} \quad k = k' \text{ and } w = w' ,\\
\mathbf{A}_{kwbl'} &= \mathbf{A}_{kwbl'} - \alpha(\mathbf{X}_{p'kwb} - \mathbf{V}_{kwbl'}) && \text{if} \quad k \neq k' ,\\
U_{kwbl'} &= U_{kwbl'} + 1 && \text{if} \quad (k = k' \text{ and } w = w') \quad \text{or} \quad k \neq k' ,
\end{aligned}$$

where l' is the index of the winning code vector in codebook $\mathbf{C}_{kwb}$ in response to the input image data $\mathbf{X}_{p'kwb}$ and α is the learning rate of the LVQ.

Step 3: Repeat steps 1 and 2 for all P training input patterns. Update all the codebooks by

$$\mathbf{V}_{kwbl} = \mathbf{V}_{kwbl} + \frac{\mathbf{A}_{kwbl}}{U_{kwbl}} \qquad \text{if} \quad U_{kwbl} \neq 0$$

for $k = 1,2,\ldots,K;\ w = 1,2,\ldots,W;\ b = 1,2,\ldots,B;\ l = 1,2,\ldots,L_{kwb}$. This forms an LVQ epoch.

Step 4: Clear **A** and **U**. Repeat steps 1 through 3 until the codebooks converge (that is, no more unintended code vectors satisfy the conditions in step 2) or a predefined number of epochs has been reached.

The neighborhood defined in step 1 for selecting the code vectors to be updated is illustrated in figure 8 for a two-dimensional vector space. The dot in the middle of this figure, labeled **X**, represents an input training pattern under consideration. The single triangle represents the group of codebooks that are dedicated to the correct aspect window and target type of this sample, with a total MSE of E^t away from **X**. The squares represent the groups of codebooks that are dedicated to other target types. One of these groups lies closest to **X**, scoring a minimum total MSE of E^*. Several other groups are also very close to **X**, and they could find a better match to **X** than the intended group. The goal of LVQ is to pull the triangle closer to **X**, while pushing the squares farther away from **X**. As we can see in step 2, no action is taken for the groups of codebooks that belong to the correct target type but are associated with wrong aspect windows (i.e., $k = k'$ and $w \neq w'$). The reason is that the wrong windows of the correct target often share certain characteristics of the correct window. Therefore, updating those codebooks in either direction may be harmful to their intended functionalities. On the other hand, there is no harm in having a wrong aspect window of the correct target type be closest to **X**, as our goal is to detect the correct target type of **X**, not its aspect.

In essence, this modified LVQ algorithm computes the appropriate decision boundary adjustments by clustering the errors caused by the adjacent code vectors. This method works well if some meaningful features have already been formed during a previous learning process, such as the K-means train-

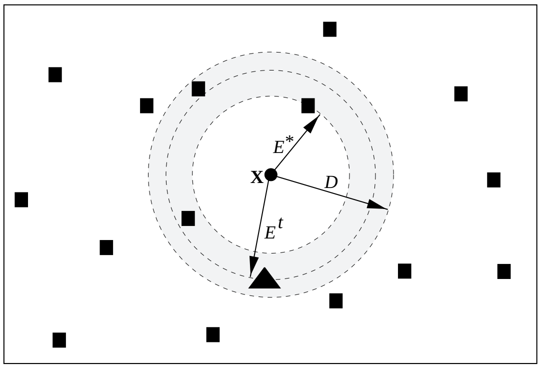

Figure 8. Neighborhood for updating procedure described for a two-dimensional vector space. Dot in center represents a training pattern. Triangle and squares represent groups of codebooks that belong to correct and wrong target class types for this input training pattern, respectively.

ing (as in this case). Otherwise, the updating neighborhood defined in step 1 may not be able to effectively avoid the adverse influence of the outliers that are embedded in the training set. Moderately large α values can be used in this method without causing stability problems, because of the averaged gradient error updates in step 3. Hence a quick and stable convergence can be achieved. As a result of boundary adjustment, an intended group of codebooks would be more likely to yield the lowest total MSE for its intended input image. Figure 9 shows all the codebooks that represent the left view

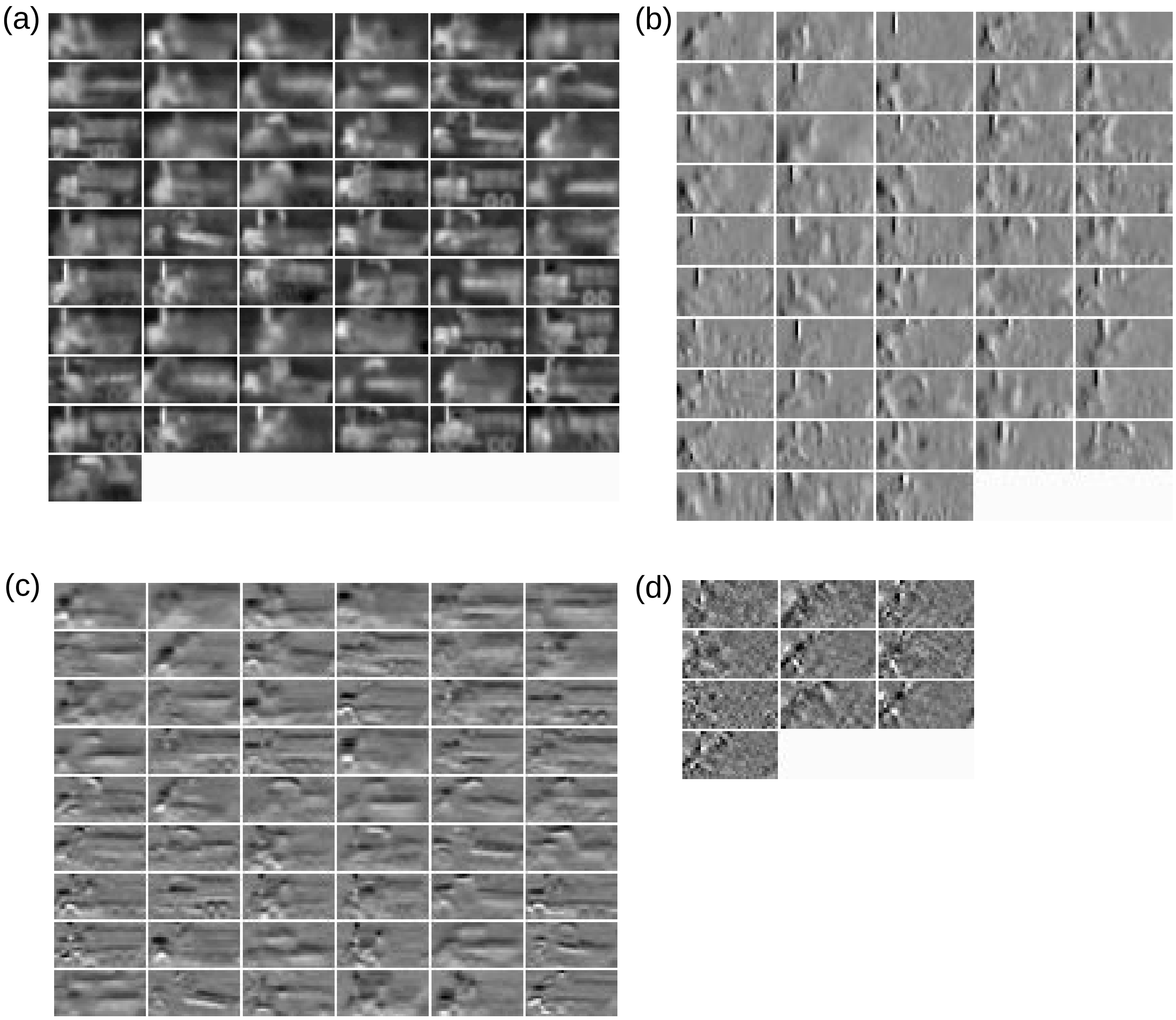

Figure 9. Content of codebooks for left view of a truck after 29 epochs of LVQ training: bands (a) LL, (b) HL, (c) LH, and (d) HH.

of a truck after 29 epochs of the LVQ training. Compared to figure 7, the features in figure 9 acquired stronger contrast and sharper edges. Apparently, the LVQ algorithm enhanced the discriminability at those regions that are critical for classification purposes. The LVQ training process also rearranged the order of code vectors based on their usage frequency and removed those code vectors with a very low usage frequency.

2.5 Processing Path Selector

Instead of passing an input image through all the processing paths available, as shown in figure 3, it is possible to use only a subset of these paths to process each input image without incurring a significant degradation in the target recognition rate. For instance, if we search only the 40 most likely paths out of 80 processing paths available in a classifier, the total computational cost could be reduced by nearly 50 percent. However, the degradation in the recognition rate may be less than 1 percent. In this way, the efficiency and response time of the proposed ATR classifier can be greatly improved. To realize this efficient scheme, we add a processing path selector at the input stage of the ATR classifier, as shown in figure 10. To avoid an excessive computational overhead, we want the path selector to be very fast and simple. Nonetheless, it must also be effective in capturing the correct path of the input image, even when only a small number of paths should be activated at any moment.

To build the path selector, we create a most *representative* image for each aspect window by taking the mean of all training images that pertain to that processing path. In other words, a Voronoi codebook with a single code vector is constructed for each processing path. As in the VQ stage, the algorithm removes the mean and variance of each input image before it is used in the path selector, so that the unwanted variations in brightness and contrast can be reduced. If we want to find the n most likely paths for an input image, we first compute and sort the MSEs between the input image and all the representative images. Then we select all the processing paths whose representative images have accounted for the n lowest MSEs. Usually the input image can find a good match with the representative image of its correct path and hence produces a relatively low MSE.

Since our path selector is a shape classifier in general, we can use the proposed LVQ algorithm to enhance its classification performance. Because there is only one code vector per processing path and no wavelet decomposition necessary, the LVQ training in this case will be much simpler and quicker than the one in the previous subsection. On the other hand, it might sometimes be difficult to produce a correct, clear-cut matching between a single representative image for a given aspect window and the input images

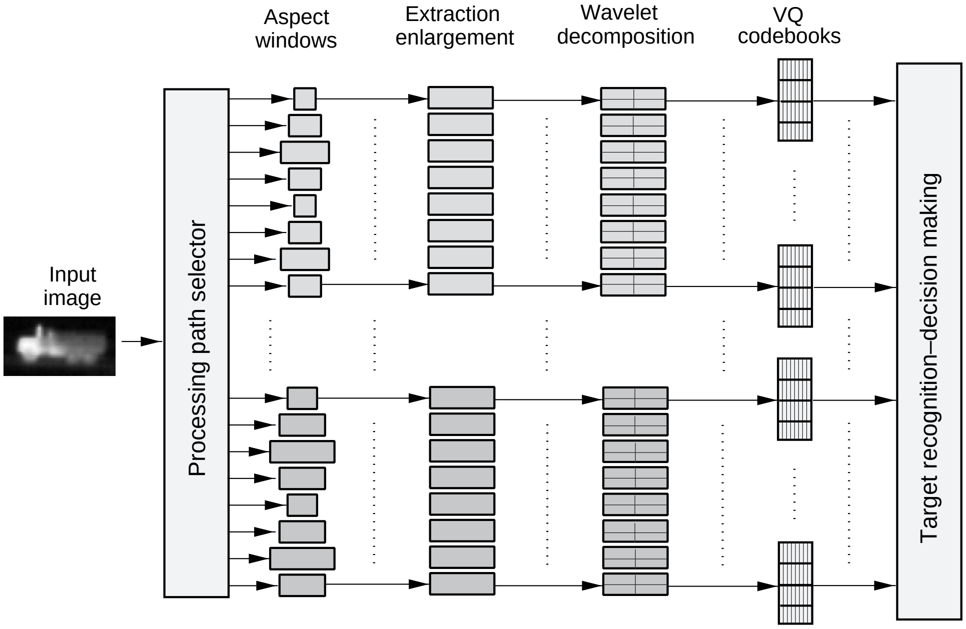

Figure 10. Proposed ATR classifier with a processing path selector.

corresponding to that aspect window. Therefore, a larger pool of candidates should be considered for updates in step 2 of our LVQ algorithm. Instead of updating only the best four candidates, the number of top candidates in the updating pool should be increased to, say, half the processing paths available in the classifier.

3. Experimental Results

To demonstrate the performance of the proposed ATR classifier, we implemented a 10-class problem. FLIR images of 10 targets were obtained at every 5° on a horizontal plane and scaled to a 2-km viewing range. The input images are 10-bit grey-scale images of size 40×75 pixels, which are assumed to have been extracted from the whole scene by an automatic cueing algorithm. For the sake of simplicity, only four aspect windows per target class (head, tail, and two sides) were created in the following experiments unless stated otherwise. The size of these aspect windows ranges from 18×29 to 29×65. The training set contains a total of 13,860 images, with 874 to 1468 images per target class. These images were taken with targets in the open and they make up the "SIG" database. On the other hand, the test set consists of 3456 images from a database called "ROI"; this set has only 5 of the 10 target classes, and there are 577 to 798 images for each of these five target classes. The ROI data were taken under less favorable conditions, such as with targets in and around clutter, with different backgrounds, and under various weather conditions; hence, these data are very challenging. Typical examples of the SIG and ROI images are shown in figure 11. These images are fed directly to the classifier without any other preprocessing or filtering.

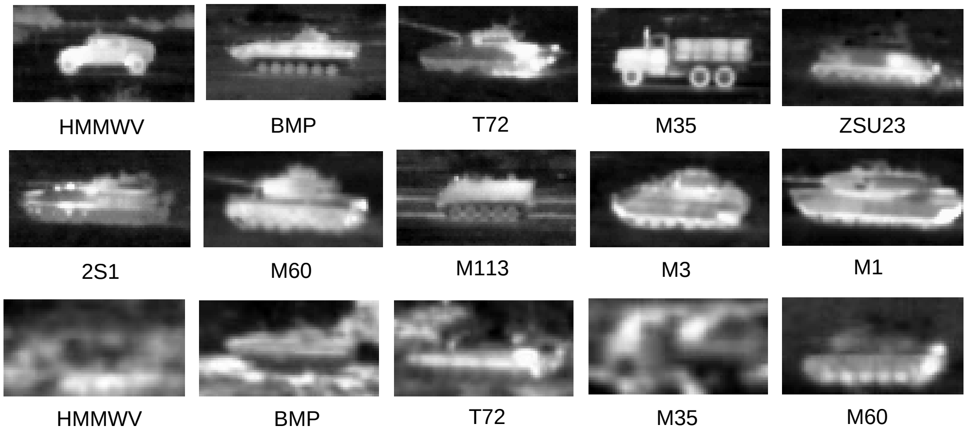

Figure 11. Examples of target types: 10 types taken from SIG database (top and center rows); these target chips are relatively easy to recognize. Last row shows five target types taken from ROI database; these images are highly cluttered and very difficult to recognize.

3.1 Proposed Method and Variants

The algorithm first decomposes each enlarged extracted area into four subbands using the Haar two-tap filter. A dedicated codebook of variable size is constructed for each of these subbands for a given aspect window. The Voronoi quantizers converge after 22 epochs of K-means training. The chance that the correct aspect window of the correct target gives the lowest total MSE (window recognition rate) is 96.05 and 63.14 percent for the SIG and ROI data, respectively. On the other hand, the target recognition rate, which is the chance of correctly identifying the target class regardless of its aspect window, is 98.11 and 69.68 percent for the SIG and ROI data, respectively.

In order to increase the discriminatory power of the classifier, we trained these Voronoi VQ further for 35 epochs with the modified LVQ algorithm proposed in this report. The E_{kw} in step 1 of the proposed LVQ algorithm was computed as the sum of the *top three* MSEs for each aspect window, so that a less reliable MSE (usually the one associated with the HH band) was ignored. In order to differentiate the usefulness of information associated with each subband, the algorithm weights the MSE produced by each subband appropriately before the top-three selection process above. The best target recognition rates achieved were 99.72 and 75.12 percent for the SIG and ROI data, respectively. The target recognition performance for each epoch of LVQ training is shown in figure 12. This figure shows that the performance of the training set saturated around 99.75 percent after 30 epochs of training, and that of the testing set deteriorated gradually after reaching its peak at the 29$^{\text{th}}$ epoch.

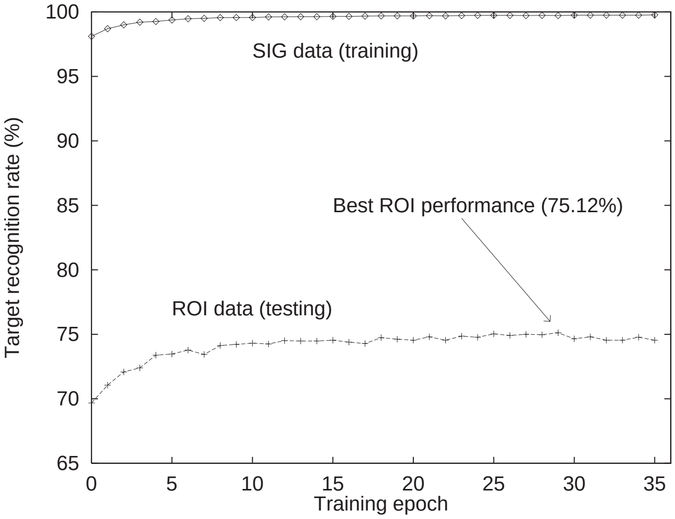

Figure 12. Target recognition performance of LVQ over epochs.

To demonstrate the superiority of the proposed method, we trained and tested three variants of this ATR classifier with the same data sets. In the first variation, the extraction enlargement stage is omitted. Different aspect windows assume different sizes for their subbands and code vectors. The total MSE is computed based on the top three subband MSEs that are normalized by the number of pixels in their corresponding subbands. We call this the *no-enlargement* variant. In the second variation, the wavelet decomposition stage is omitted. The whole enlarged extraction area is used to build a single codebook for each aspect window. Hence the size of all code vectors is the same as the size of the enlarged extraction. We refer to this procedure as the *one-band* variant. Finally, the *joint-band* variant concatenates all the four subbands together and then builds a single codebook for these concatenated bands; hence, the advantages of product code VQ no longer exist.

Table 1 shows the best performances of the K-means and the LVQ training of these three variants, together with the proposed method. The proposed method clearly outperformed all three variants in terms of recognition performance and generalization capability. The performance of the no-enlargement variant is the closest to the proposed method, trailing by just about 1 percent in all categories. Omitting the enlargement stage and performing comparison with smaller code vectors, this variant is computationally more efficient than the proposed method. Therefore, the no-enlargement variant might be used when the efficiency of recognition is very critical. On the other hand, the joint-band variant required almost the same amount of computational resources as the proposed method, but its performance is significantly worse than that of the proposed method. Without the advantages of product code VQ, the joint-band variant is less useful in any situation. Finally, being deprived of the benefits of wavelet decomposition and product code VQ, the one-band variant performed the worst in all categories. Comparing the one-band variant to the joint-band variant, we can see that the wavelet decomposition alone has accounted for a 7.15-percent difference in the test performance.

Table 1. Best window and target recognition rates after K-means and LVQ training achieved by proposed method and its three variants. (Best rates in bold.)

Training	Data	Recognition rate for various methods (%)							
		Proposed		No enlargement		One band		Joint band	
		Window	Target	Window	Target	Window	Target	Window	Target
K-means	SIG	96.05	98.11	95.81	98.00	88.77	92.60	92.11	95.18
	ROI	63.14	69.68	61.43	69.16	40.80	47.92	55.06	61.55
LVQ	SIG	98.22	**99.72**	98.06	99.70	94.20	97.60	95.58	98.36
	ROI	70.43	**75.12**	68.61	73.90	56.28	62.53	63.89	69.68

3.2 Data and Aspect Windows

In previous experiments, the test performance on the ROI database was significantly lower than on the SIG database. These results suggest that the ROI database may contain many unique characteristics that are too difficult to capture from the samples of the SIG data. Hence we formed a new training set by randomly selecting 80 percent of the images from both the SIG and ROI databases. The remaining 20 percent of images in both databases constituted a new testing set. We retrained the proposed ATR classifier with the new training set.

As shown in the fourth column of table 2, the target recognition rates after the K-means training were 96.89 and 88.88 percent for the training and testing set, respectively. With LVQ training, the best target recognition rates went up to 99.47 and 93.53 percent for the training and testing sets, respectively. Compared to the best results in table 1, the target recognition rate after LVQ training improved from 75.12 to 93.53 percent with the new test set, while the recognition rate on the training set remained almost the same. Of the ROI images in the new test set, 91.03 percent were correctly recognized. Therefore, after learning some characteristics of the ROI data, the classifier can now perform almost as accurately with the ROI as with the SIG test images.

In all the experiments discussed so far, four aspect windows were created for each target type. We also investigated the effect of increasing the number of aspect windows on the recognition performance of the classifier. The classifier was reconfigured so that eight aspect windows were created for each target class, as illustrated in figure 4. The new data sets just described were used to train and test this new configuration. After the K-means training, the target recognition rates were 96.84 and 88.45 percent for the training and testing sets, respectively. After the LVQ training, the corresponding rates were raised to 99.28 and 93.01 percent, respectively. Compared with the best target recognition rates in the four-window configuration, there was

Table 2. Performance of proposed ATR classifier configured with four and eight aspect windows, respectively. New training and testing sets that consist of both SIG and ROI target chips were used. (Best results in bold.)

Training	Data	Performance with different number of aspect windows (%)			
		Four aspect windows		Eight aspect windows	
		Window	Target	Window	Target
K-means	Train	94.88	96.89	93.54	96.84
	Test	82.96	88.88	78.52	88.45
LVQ	Train	98.12	**99.47**	96.69	99.28
	Test	87.58	**93.53**	82.41	93.01

about 0.19 and 0.52 percent degradation in the training and testing sets, respectively. The number of processing paths was doubled from 40 to 80 in this case, but the average codebook size was nearly halved, from 30.4 to 16.6 code vectors.

Therefore, adding oblique windows did not bring the expected improvement in performance, while the amount of computation was increased by about 9 percent, based on the total number of code vectors processed per input image. One possible reason for the slightly poorer performance is the reduction of the training data per codebook during the formation of Voronoi quantizers. In addition, the oblique windows, having the smallest range of aspects (a range of 25°), may not have captured features embedded in their inputs that were sufficiently distinctive for the input images to be correctly identified.

3.3 Processing Path Selector

We tested the processing path selector with both the four- and eight-window configurations. In both cases, the best codebook array obtained after the LVQ training was used in the VQ stage. We set the path selector so that the top 1/4, 1/2, 3/4, or all of the available processing paths were used for a given input image. The target recognition rates obtained with different sets of processing paths activated are given in table 3 for both the four- and eight-window configurations.

With only a subset of processing paths activated, we can see that the degradation in performance is indeed very small. For instance, with only 20 out of the 80 paths activated in the eight-window configuration, the test performance decreased by merely 3.35 percent. This degradation was further reduced to 0.78 percent when 40 out of the 80 paths were used. It is interesting to observe that with half or less of the processing paths activated, the eight-window configuration outperformed the four-window configuration. Therefore, the eight-window configuration could be a better alternative than the four-window configuration in situations where operational efficiency is a real concern.

Table 3. Target recognition rate achieved when different sets of processing paths are chosen by a path selector.

Configuration	Data	Performance with different sets of paths used (%)			
		Top $\frac{1}{4}$	Top $\frac{1}{2}$	Top $\frac{3}{4}$	All paths
Four windows	Train	91.45	96.93	98.87	99.47
(40 paths)	Test	86.89	91.94	92.95	93.53
Eight windows	Train	94.72	98.09	98.99	99.28
(80 paths)	Test	89.66	92.23	92.93	93.01

4. Comparison with Other Results

Considering the amount of training and test data used in our recognition problem, the results obtained in our experiments are quite good. Since there is neither a standard set of ATR system requirements nor a common FLIR testing set for ATR problems, it is relatively difficult to compare our results with most existing ATR classifiers. For most ATR classifiers described in the literature, the authors generally considered only a few target classes (five or fewer) and used very small training and testing sets (fewer than 100 samples).

Fortunately, we can make a suitable comparison with the results published by Mirelli and Rizvi [48], because we and they used identical SIG and ROI databases. (No other published results use the same databases.) The schematic diagram of Mirelli and Rizvi's complete ATR classification system is shown as figure 13. They used a group classifier to determine the general shape of an incoming target, followed by 14 similar multilayer convolution neural networks (MLCNNs) that were optimized to recognize targets of particular size and shape. Each MLCNN consists of four layers that are made up of 25 to 50 convolutional kernels, so that each is a rather complex module by itself. The outputs of these MLCNNs were combined by another MLP to produce the final recognition score. In a 10-class problem, the best recognition result Mirelli and Rizvi obtained for the complete ROI testing set was 73.41 percent, which is slightly lower than the 75.12 percent that our classifier obtained under the same conditions. As they reported a 2.2-percent improvement brought by the final MLP alone, we expect that our result can also be further improved by the addition of an appropriate postprocessor at the end of our processing paths.

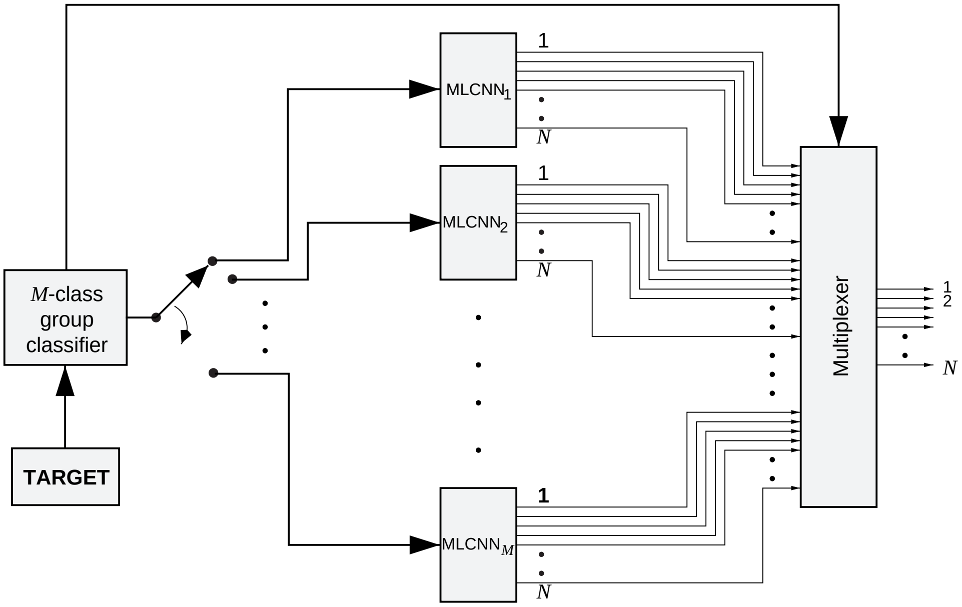

Figure 13. ATR classifier proposed by Mirelli and Rizvi [48].

5. Conclusions

A sad lesson learned by our armed forces during Operation Desert Storm is the difficulty of distinguishing between friendly and hostile vehicles or equipment under poor visibility conditions. Consequently, more than half of the American casualties were inflicted by so-called "friendly fire." To overcome similar problems in the future, we need ATR systems that are effective and efficient in distinguishing enemy and friendly targets under adverse environmental conditions. For this reason, a number of ATR research projects are actively carried out by the U.S. Army Research Laboratory. In the long run, these efforts could significantly contribute to the safety of our troops and the security of our nation.

In this report, we propose an ATR classifier that consists of several aspect windows, an extraction enlargement procedure, a wavelet decomposition, and a set of product code VQs. Background noise and effective dimensionality are greatly reduced by the variable-size aspect windows. As shown by the performance of the no-enlargement method, the extraction enlargement is necessary to achieve higher recognition rates with a more compatible similarity measure between the aspect windows. Wavelet decomposition of the enlarged extraction subdivides the complexity of the recognition task, extracts target features at different perspectives, and enables the use of a product code VQ. The K-means algorithm and a modified LVQ algorithm have been used to capture intra-target similarities and to enhance inter-target discriminability. We tested three variants of the proposed ATR classifier in order to examine the effects of the extraction enlargement procedure, the wavelet decomposition, and the product code VQ. Each of these three procedures was one component of the proposed method. We constructed each variant by omitting one of the three procedures. It was found that all the three variants resulted in recognition performance that was worse than the proposed method; therefore, each of the stages is critical to the proposed method.

References

1. D. C. Carmer and L. M. Peterson, “Laser Radar in Robotics,” *Proc. IEEE* **84**, no. 2, 299–320, 1996.

2. R. Chellappa, C. L. Wilson, and S. Sirohey, “Human and Machine Recognition of Faces: A Survey,” *Proc. IEEE* **83**, no. 5, 705–740, 1995.

3. A. Samal and P. A. Iyengar, “Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey,” *Pattern Recognition* **25**, no. 1, 65–77, 1992.

4. C. C. Tappert, C. Y. Suen, and T. Wakahara, “The State of the Art in On-line Handwriting Recognition,” *IEEE Trans. Pattern Anal. Machine Intell.* **12**, no. 8, 787–808, 1990.

5. R. G. Casey and E. Lecolinet, “A Survey of Methods and Strategies in Character Segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.* **18**, no. 7, 690–706, 1996.

6. B. Dasarathy, “Information Processing for Target Recognition from Autonomous Vehicles,” *Proc. SPIE Electro-Optical Tech. Autonomous Vehicles* **219**, 86–93, 1980.

7. G. R. Osche and D. S. Young, “Imaging Laser Radar in the Near and Far Infrared,” *Proc. IEEE* **84**, no. 2, 103–125, 1996.

8. C. M. Lo, “Forward-Looking Infrared (FLIR) Image Enhancement for the Automatic Target Cuer System,” *Proc. SPIE Image Processing for Missile Guidance* **238**, 91–102, 1980.

9. J. G. Verly and R. L. Delanoy, “Model-Based Automatic Target Recognition System for Forward-Looking Ground-Based and Airborne Imaging Laser Radars,” *Proc. IEEE* **84**, no. 2, 126–163, 1996.

10. A. P. Kramer, D. Perschbacher, R. Johnston, and T. Kipp, “Relational Template Matching Algorithm for FLIR Automatic Target Recognition,” *Proc. SPIE Architecture, Hardware, and FLIR Issues in ATR* **1957**, 29–37, 1993.

11. R. J. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*, New York: Wiley, 1992.

12. Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” *IEEE Trans. Pattern Anal. Machine Intell.* **17**, no. 8, 790–799, 1995.

13. R. D. Short and K. Fukunaga, “The Optimal Distance Measure for Nearest Neighbor Classification,” *IEEE Trans. Inform. Theory* **27**, 622–627, 1981.

14. S. Salzberg, A. L. Delcher, D. Heath, and S. Kasif, “Best-Case Results for Nearest-Neighbor Learning,” *IEEE Trans. Pattern Anal. Machine Intell.* **17**, no. 6, 599–608, 1995.

15. T. Kohonen, “Improved Versions of Learning Vector Quantization,” in *Proc. Int. Joint Conf. Neural Networks* **1**, 545–550, San Diego, 1990.

16. T. Kohonen, *Self-Organizing Maps*, Heidelberg: Springer Series in Information Sciences **30**, 1995.

17. R. P. Lippmann, “Pattern Classification using Neural Networks,” *IEEE Commun. Magazine*, 47–64, November 1989.

18. G. A. Carpenter, “Neural Network Models for Pattern Recognition and Associative Memory,” *Neural Networks* **2**, 243–257, 1989.

19. G. A. Carpenter, S. Grossberg, and C. Mehanian, “Invariant Recognition of Cluttered Scenes by a Self-Organizing ART Architecture: CORT-X Boundary Segmentation,” *Neural Networks* **2**, 169–181, 1989.

20. L. Spirkovska and M. B. Reid, “Coarse-Coded Higher-Order Neural Networks for PSRI Object Recognition,” *IEEE Trans. Neural Networks* **4**, no. 2, 276–283, 1993.

21. I. E. Dror, M. Zagaeski, and C. F. Moss, “Three-Dimensional Target Recognition via Sonar: A Neural Network Model,” *Neural Networks* **8**, no. 1, 149–160, 1995.

22. A. Ravichandran and B. Yegnanarayana, “Studies on Object Recognition from Degraded Images using Neural Networks,” *Neural Networks* **8**, no. 3, 481–488, 1995.

23. P. Sajda, C. D. Spence, S. Hsu, and J. C. Pearson, “Integrating Neural Networks with Image Pyramids to Learn Target Context,” *Neural Networks* **8**, no. 7/8, 1143–1152, 1995.

24. L. A. Chan, N. M. Nasrabadi, and V. Mirelli, “Automatic Target Recognition Using Modularly Cascaded Vector Quantizers and Multilayer Perceptrons,” *Proc. Int. Conf. Acoust. Speech Signal Process.* **6**, 3387–3390, Atlanta, 1996.

25. L. A. Chan, N. M. Nasrabadi, and V. Mirelli, “Multi-stage Target Recognition using Modular Vector Quantizers and Multilayer Perceptrons,” *Proc. Computer Vision Pattern Recognition* **1**, 114–119, San Francisco, 1996.

26. G. F. McLean, “Vector Quantization for Texture Classification,” *IEEE Trans. Syst. Man Cybern.* **23**, no. 3, 637–649, 1993.

27. A. J. O’Toole, H. Abdi, K. A. Deffenbacher, and D. Valentin, “Low-Dimensional Representation of Faces in Higher Dimensions of the Face Space,” *J. Opt. Soc. Am.* **10**, no. 3, 405–411, 1993.

28. D. L. Swets and J. Weng, “Using Discriminant Eigenfeatures for Image Retrieval,” *IEEE Trans. Pattern Anal. Machine Intell.* **18**, no. 8, 831–836, 1996.

29. N. Hess-Nielsen and M. V. Wickerhauser, “Wavelets and Time-Frequency Analysis,” *Proc. IEEE* **84**, no. 4, 523–540, 1996.

30. R. Anand, K. Mehrotra, C. K. Mohan, and S. Ranka, “Efficient Classification for Multiclass Problems Using Modular Neural Networks,” *IEEE Trans. Neural Networks* **6**, no. 1, 117–124, 1995.

31. W. E. Alexander, D. S. Reeves, and C. S. Gloster, Jr., “Parallel Image Processing with the Block Data Parallel Architecture,” *Proc. IEEE* **84**, no. 7, 947–968, 1996.

32. A. Krikelis and R. M. Lea, “A Modular Massively Parallel Computing Approach to Image-Related Processing,” *Proc. IEEE* **84**, no. 7, 988–1004, 1996.

33. D. W. Hammerstrom and D. P. Lulich, “Image Processing Using One-Dimensional Processor Arrays,” *Proc. IEEE* **84**, no. 7, 1005–1018, 1996.

34. O. Rioul and M. Vetterli, “Wavelets and Signal Processing,” *IEEE Signal Process. Magazine*, 14–37, October 1991.

35. M. Unser and M. Eden, “Multi-resolution Feature Extraction and Selection for Texture Segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.* **11**, no. 7, 717–728, 1989.

36. C. H. Chen and G. G. Lee, “Multi-resolution Wavelet Analysis Based Feature Extraction for Neural Network Classification,” *Proc. Int. Conf. Neural Networks* **3**, 1416–1421, June 1996.

37. J. Woods and S. O’Neil, “Subband Coding of Images,” *IEEE Trans. Acoust. Speech Signal Process.* **34**, no. 5, 1278–1288, 1986.

38. M. Antonini, M. Barlaud, P. Mathiew, and I. Daubechies, “Image Coding Using Wavelet Transform,” *IEEE Trans. Image Process.* **1**, no. 2, 205–220, 1992.

39. P. Sriram and M. V. Marcellin, “Image Coding Using Wavelet Transform and Entropy-Constrained Trellis-Coded Quantization,” *IEEE Trans. Image Process.* **4**, no. 6, 725–733, 1995.

40. S. G. Mallat, “Wavelets for a Vision,” *Proc. IEEE* **84**, no. 4, 604–614, 1996.

41. A. Graps, “An Introduction to Wavelets,” *IEEE Comput. Sci. Eng.* **2**, no. 2, 1995.

42. A. Croisier, D. Esteban, and C. Galand, “Perfect Channel Splitting by Use of Interpolation/Decimation/Tree Decomposition Techniques,” *Proc. IEEE Int. Conf. Inform. Sci. Syst.*, Patras, Greece, 1976.

43. M. Vetterli and C. Herley, “Wavelets and Filter Banks: Theory and Design,” *IEEE Trans. Signal Process.* **40**, no. 9, 2207–2232, 1992.

44. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, New Jersey: Prentice Hall, 1995.

45. J. D. Villasenor, B. Belzer, and J. Liao, “Wavelet Filter Evaluation for Image Compression,” *IEEE Trans. Image Process.* **4**, no. 8, 1053–1060, 1995.

46. A. Gersho and R. M. Gray, “Vector Quantization and Signal Compression,” Boston: Kluwer Academic Publishers, 1992.

47. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

48. V. Mirelli and S. A. Rizvi, "Automatic Target Recognition Using a Multilayer Convolution Neural Network," *Proc. SPIE Signal Process. Sensor Fusion Target Recog.* **2755**, 106–126, 1996.

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT) G Singley
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

OSD
Attn OUSD(A&T)/ODDDR&E(R) J Lupo
Washington DC 20301-7100

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

CECOM RDEC Elect System Div Dir
Attn J Niemela
FT Monmouth NJ 07703

CECOM
Sp & Terrestrial Commctn Div
Attn AMSEL-RD-ST-MC-M H Soicher
FT Monmouth NJ 07703-5203

Dept of the Army (OASA) RDA
Attn SARD-PT R Saunders
103 Army
Washington DC 20301-0103

Dir of Assessment and Eval
Attn SARD-ZD H K Fallin Jr
103 Army Pentagon Rm 2E673
Washington DC 20301-0163

Dpty Assist Secy for Rsrch & Techl
Attn SARD-TT F Milton Rm 3EA79
The Pentagon
Washington DC 20301-0103

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

MICOM RDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

US Army Avn Rsrch, Dev, & Engrg Ctr
Attn T L House
4300 Goodfellow Blvd
St Louis MO 63120-1798

US Army CECOM
Night Vision & Elec Sensors Dir
Attn AMSEL-RD-NV-VISPD C Hoover
Attn B Redman
Attn D N Barr
10221 Burbeck Rd, Ste 430
FT Belvoir VA 22060-5806

US Army CECOM Rsrch, Dev, & Engrg
Attn R F Giordano
FT Monmouth NJ 07703-5201

US Army Edgewood Rsrch, Dev, & Engrg Ctr
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Matl Cmnd
Dpty CG for RDE Hdqtrs
Attn AMCRD BG Beauchamp
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd
Prin Dpty for Acquisition Hdqrts
Attn AMCDCG-A D Adams
5001 Eisenhower Ave
Alexandria VA 22333-0001

Distribution

US Army Matl Cmnd
Prin Dpty for Techlgy Hdqrts
Attn AMCDCG-T M Fisette
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Natick Rsrch, Dev, & Engrg Ctr
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

US Army NVESD
Attn AMSRL-RD-NV-UAB C Walters
10221 Burbeck Rd Ste 40
FT Belvoir VA 22060

US Army Rsrch Ofc
Attn G Iafrate
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn
Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv & Armaments Cmnd
Attn AMSTA-AR-TD C Spinelli
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Tank-Automtv Cmnd Rsrch, Dev, &
Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Test & Eval Cmnd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Army Train & Doctrine Cmnd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Dept of Mathematical Sci
Attn MAJ D Engen
West Point NY 10996

USAASA
Attn MOAS-AI W Parron
9325 Gunston Rd Ste N319
FT Belvoir VA 22060-5582

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

GPS Joint Prog Ofc Dir
Attn COL J Clay
2435 Vela Way Ste 1613
Los Angeles AFB CA 90245-5500

Special Assist to the Wing Cmndr
Attn 50SW/CCX Capt P H Bernstein
300 O'Malley Ave Ste 20
Falcon AFB CO 80912-3020

DARPA
Attn B Kaspar
Attn L Stotts
3701 N Fairfax Dr
Arlington VA 22203-1714

ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
University of Texas
Austin TX 78712

Univ of Maryland Dept of Elec Engrg
Attn Q Zheng
Attn R Chellappa
A V Williams Bldg Rm 2365
College Park MD 20742-3285

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

ERIM
Attn C Dwan
Attn D Zhao
Attn J Ackenhusen
Attn Q Holmes
1975 Green Rd
Ann Arbor MI 48105

Distribution

Lockheed Martin Vought
Attn B Evans Stop L37-01
Attn D DuBois Stop LOST-20
Attn K Jenkins Stop L37-01
Dallas TX 7526-0003

Palisades Instit for Rsrch Svc Inc
Attn E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

SandersLockheed Martin Co
Attn PTP2-A001 K Damour
PO Box 868
Nashua NH 03061-0868

US Army Rsrch Lab
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records
  Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)

US Army Rsrch Lab (cont'd)
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-EE Z G Sztankay
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE H Kwon
Attn AMSRL-SE-SE H Moon
Attn AMSRL-SE-SE J Phillips
Attn AMSRL-SE-SE L A Chan (5 copies)
Attn AMSRL-SE-SE L Bennett
Attn AMSRL-SE-SE L-C Wang
Attn AMSRL-SE-SE M Venkatraman
Attn AMSRL-SE-SE M Vrabel
Attn AMSRL-SE-SE N Nasrabadi
Attn AMSRL-SE-SE P Rauss
Attn AMSRL-SE-SE S Der
Attn AMSRL-SE-SE T Kipp
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1997

3. REPORT TYPE AND DATES COVERED
Interim, from December 1996 to July 1997

4. TITLE AND SUBTITLE
Automatic Target Recognition Using Wavelet-Based Vector Quantization

5. FUNDING NUMBERS
PE: 61102A

6. AUTHOR(S)
Lipchen Alex Chan and Nasser M. Nasrabadi

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

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1503

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
AMS code: 611102.305
ARL PR: 7NE0M1

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An automatic target recognition classifier is described that uses a set of dedicated vector quantizers (VQs) in the wavelet domain. The background pixels in each input image are properly clipped out by a set of aspect windows. The extracted target area for each aspect window is then enlarged to a fixed size, after which a wavelet decomposition is used to split this region into several subbands. A dedicated VQ codebook is then generated for each subband of a particular target class at a specific range of aspects. Thus, each codebook consists of a set of feature templates that are iteratively adapted to represent a particular subband of a given target class at a specific range of aspects. These templates are then further trained by a modified learning vector quantization (LVQ) algorithm that enhances their discriminatory characteristics. Finally, a path selector was designed to speed up the recognition process at the expense of a tolerable degradation in the recognition rate.

14. SUBJECT TERMS
Target recognition, vector quantization, wavelet, FLIR imagery

15. NUMBER OF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

DEPARTMENT OF THE ARMY
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197