CHAPTER III

MULTIPLE DESCRIPTION CODING FOR MULTIPLE CLASSIFIER SYSTEMS

In this chapter, a new method of combining multiple classifiers is presented with the goal of improving the classification accuracy of ECOC and other combining algorithms. This approach employs a *deterministic* redundancy between classifiers through the use of multiple description (MD) coding models. In contrast, most of the traditional combining algorithms (e.g., Bagging, Adaboost and random subspace method) usually manipulate the *statistical* redundancies between classifiers. In particular, a multiple classifier system is constructed using a new and efficient wavelet-based MD pattern analysis algorithm.

Basically, linear transforms and expansions are the fundamental mathematical tools of signal processing, yet the properties of linear expansions for preprocessing of data multiple classifier systems are not fully understood. In Section 3.2, we describe wavelet based signal processing, called *local discriminant bases* that is suitable for classification problems. In fact, this technique has been successfully used as a feature extraction for a single classification rule. Section 3.3 is devoted to multiple description coding models with overcomplete wavelet representation. The overcomplete wavelet representation is based on frame of the shift-variant wavelet discrete wavelet transform. Then in Sections 3.4 and 3.5, we discuss how to construct independent and unequal error protected descriptions for multiple classifiers. In particular, we propose a pattern recognition system that consists of a set of independent classifiers which use overcomplete feature descriptions obtained from a set of collections of bases (or equivalently selected from a set of overcomplete libraries of local discriminant bases). In this system, the classification procedures is in parallel.

3.1 Introduction

The problems of breaking an image into pieces and then being able to reconstruct it from an arbitrary subset of these pieces have been long discussed, i.e., the problem of source coding [44] and optical holography [53]. In source coding, the proposition of jointly coding of many source descriptions is based on the original questions posed by Gersho, Ozarow, Witsenhausen, Wolf, Wyner, and Ziv at the Shannon Theory Workshop in the September 1979 (see [54] and the references therein). The question was that *if an information source is described by two separate descriptions, what are the concurrent limitations on qualities of these descriptions taken separately and jointly?* This problem would come to be known as the *multiple description problem*.

Multiple description (MD) coding is a source coding technique, in which the source is encoded into multiple descriptions, which are transmitted over different channels to the receiver. When the succeeding descriptions are available to the receiver, they can be used to refine the information contained in the preceding descriptions.

There are several approaches proposed for constructing multiple description coding, and one of the approaches is *multiple description transform coding*. Multiple description coding with frame expansions [44] is one of the successful attempts that employs the frame decomposition approach for MD transform coding. In multiple description coding with frame expansions, the source vector is expanded using a linear redundant transform. The expanded signal has a linear dependent between its components. This clearly corresponds to a frame decomposition. After expanding the signal, each redundant coefficient is quantized, and the quantized codewords are sent on the channel.

In the conventional communication system based on error correcting codes, each component of the source vector is quantized using a quantization codebook. A linear block code is then applied to the quantizer output, and the encoding codewords are sent on the channel. Due to the fact that a linear block code is a linear transform, the difference between the systems is the swapping of transform and quantization operations.

As discussed in [8], each concept belonging to the class information source can be represented by a quantized bit stream. Under the theory of covering numbers (Kolmogorov's entropy) [40, 41], the quantized bit stream is considered to be quantized using a quantization codebook learned from the classification data. Drawing an analogy between classification and source–channel models, a linear block code can then be applied to the classification quantizer output in ECOC with success. Contrary to ECOC, in our proposed scheme, the class information source is expanded using a linear redundant transform. Then, each of the expanded class information is conceptually quantized using a quantization codebook learned from the classification data. It is in this context that the swapping of transform and quantization operations in classification is understudied, especially in the exploitation of frame expansions for multiple classifier systems.

Drawing an analogy between classification and source–channel models, classification with k classifiers is then equivalent to generalized MD coding with k channels. Therefore, assimilating the MD coding with frame expansions to new multiple classifier systems is closely related to the assimilation of the error correcting codes to ECOC, but is much more efficient.

3.2 A Library of Orthonormal Bases and Local Discriminant Basis

Standard discrete wavelet packet transform (DWPT) is a generalization of discrete wavelet transform (DWT) that offers a richer range of possibilities for signal and image representations. Local discriminant basis (LDB) [55] is a generalization of DWPT that is implemented for selecting optimal local basis using class separability criteria.

There are several advantages in using LDB for feature extractors. For example, LDB is more robust to outliers and perturbations than the Karhunen-Loéve transform and the linear discriminant analysis. Moreover, LDB can capture local features with less computation and possesses structural interpretability. It is also reported that more resistance to overtraining is obtained when we use classifier with LDB. However, *Coiflets* seem to be less resistant to overtraining than other wavelet filters, as they are adapted too well to training data [15]. Adding redundancy to LDB is an improvement technique proposed in Section 3.4 in order to reduce the overtraining problem retained in LDB method, inspiring from the fundamental concept used in multiple classifier systems.

Note that in this section, only relevant material to the one–dimensional (1D) wavelet transforms and its variants will be presented. An extension to 2D is easily obtained by using tensor products of the transforms.

3.2.1 Wavelet Bases, Wavelet Packet Bases and Best Basis Selection

Both DWT and DWPT can be described and computed by a pair of quadrature mirror filters (QMF) H and G. The filter H is a lowpass filter with a finite impulse response denoted by $h(n)$ of length K. The detail of various design criteria for the lowpass filter coefficients $h(n)$ can be found in the literature [56–58]. Once $h(n)$ is derived, we can have the highpass filter G with a finite impulse response defined by $g(n) = (-1)^n h(1-n)$, for a finite length $h(n)$. These filters are called QMF, if they satisfy the following orthogonality (or perfect reconstruction) conditions:

$$HG^* = GH^* = 0 \text{ and } H^*H + G^*G = I, \tag{3.1}$$

where H^* and G^* are adjoint operations of H and G, and I is an identity operator, respectively.

In DWT analysis, the signal is split into approximation and detail subbands defined by two subsequences $H\mathbf{x}$ and $G\mathbf{x}$ of lengths $N/2$, where $\mathbf{x}$ is the signal vector of length N. The approximation subband is then itself split into second-level approximation and detail subbands, and the process is recursively repeated.

In DWPT analysis, the detail subbands as well as the approximation subbands can also be split. For example, the first level decomposition generates $H\mathbf{x}$ and $G\mathbf{x}$ just likes in the DWT. The second level decomposition generates four subbands, $H^2\mathbf{x}, GH\mathbf{x}, HG\mathbf{x}, G^2\mathbf{x}$. And so on, the L level decomposition generates 2^L subbands. Because of the perfect reconstruction condition on H and G, each decomposition step is considered as a decomposition of the vector space into mutually orthogonal subspaces. With the application of the projection operators H and G to the parent subspaces at each decomposition level, wavelet packet transform naturally generates subspaces of $\mathbf{R}^N$ of a binary tree, where the nodes of the tree represent subspaces with different time–frequency localization characteristics. This leads to an indication that each representation subspace is spanned by a set of basis vectors, so called *wavelet packets*. In the general case of L level decomposition, we have 2^L and 4^L

possible ways to represent signal and image by using a redundant set of wavelet packets in the binary (quad) tree, respectively. Clearly, an extremely large amount of freedom exists for the construction of orthogonal bases from the wavelet packet library. This greater flexibility is exploited to increase the efficiency of the representation.

Because the collection of wavelet packets is overcomplete, Coifman and Wickerhauser [59] suggested to use a fast dynamic programming algorithm to search wavelet packets for that *best basis* which is optimal according to a given cost function $\mathcal{M}$. The algorithm suggested by Coifman is called the *best basis* algorithm.

3.2.2 Local Discriminant Basis

Coifman and Saito [55, 60] extended the "best basis" method to select an orthonormal basis suitable for signal/image classification problems. In particular, they suggested to use a variety of cost functions that measure the class separability among classes of signal/image patterns. More precisely, let $\mathbf{z} = \{z_i\}_{i=0}^{N-1}$ be a sequence, e.g., decomposition vector. And, let $\mathbf{p} = \{p_i\}_{i=0}^{N-1}$ and $\mathbf{q} = \{q_i\}_{i=0}^{N-1}$ be two nonnegative sequences with $\sum p_i = \sum q_i = 1$. For this purpose, the selection criterion (cost function) is in general an entropy criterion $\mathcal{M}$ suitable for signal/image compression like the non-normalized Shannon entropy,

$$\mathcal{M}(\mathbf{z}) = -\sum_{i=0}^{N-1} |z_i|^2/||\mathbf{z}||^2 \log |z_i|^2/||\mathbf{z}||^2, \tag{3.2}$$

has to be replaced by a discriminant information function $\mathcal{D}(\mathbf{p}, \mathbf{q})$ [55, 60].

In the two–class case, the discriminant information function $\mathcal{D}(\mathbf{p}, \mathbf{q})$ measures how differently $\mathbf{p}$ and $\mathbf{q}$ are distributed. One possibility of the measure $\mathcal{D}$ is the relative entropy of two sequences defined by

$$\mathcal{W}(\mathbf{p}, \mathbf{q}) \equiv \sum_{i=0}^{N-1} (p_i - q_i)^2. \tag{3.3}$$

In this work, only the $\mathcal{D} = \mathcal{W}$ (see other discriminant information functions in [55]) is used. In the general case of C classes, the discriminant measure of C sequences is defined as

$$\mathcal{D}(\{\mathbf{p}^{(c)}\}_{c=1}^{C}) = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} \mathcal{D}(\mathbf{p}^{(i)}, \mathbf{q}^{(j)}). \tag{3.4}$$

Note that the discriminant measure should be *additive* in order to ensure a fast computational algorithm.

Given an additive discriminant measure, we are capable of evaluating the power of discrimination of each basis vector in the wavelet packet library. We are now ready to describe the *local discriminant basis* algorithm. First, we decompose training signal data from different classes using a collection of wavelet packets, and form them into a binary tree. Then, we compute the *time–frequency energy map* for each class by accumulating

the energy of expansion coefficients of the signal at each position in the tree followed by the normalization by the total energy of the signals belonging to class $c \in C$. Next, we compute the discriminant measure at each position in the tree by letting $\mathbf{p}$ and $\mathbf{q}$ in (3.3) and (3.4) be the time–frequency energy map sequences between different classes. Similar to the best–basis algorithm, we search for the optimal basis according to the criterion defined by the derived discriminant measures in order that no other basis in the library will discriminate more between classes. Initially, local discriminant basis (LDB) is set to be the basis vectors of the children nodes at the bottom of the binary tree. Then, the discriminant measure functions of each two children nodes are compared to their parent's. If the sum of the discriminant measure functions of two children nodes is higher than their parent, we keep the basis vectors of the children nodes. Otherwise, the basis vectors of the parent node is chosen as LDB. We repeat the comparisons until we reach at the top of the tree. After this step, we have a complete orthonormal basis LDB.

Instead of using a complete orthonormal basis LDB as features, we consider the case that only a few *good* features are extracted from LDB and used to train classifiers. Since the dimensionality of the problem is reduced, it is more likely that both the accuracy and speed of the trained classifiers can be improved. Once the LDB is selected, we evaluate the power of discriminant of each individual basis function in the LDB by computing the discriminant measure in (3.4) using the time–frequency energy maps of expansion coefficients of signals obtained from a single basis function in the LDB belonging to class $c \in C$, instead of a set of basis vectors at that position in the tree. Then, we sort the basis vectors of the LDB in the order of their discriminant power. Next, we retain the first M basis vectors as the *most discriminant basis* (MDB) functions. Throughout the rest of this paper, the MDB functions will be referred to as the M most important LDB vectors, and these two terms can be used interchangeably whenever it is appropriated.

3.3 Multiple Description Coding Models

In this section, we first describe the *frame*. Frame and overcomplete representation are two key element for adapting shift-variant wavelet transform for multiple description coding models. In fact, they can be interchangeable whenever they are appropriated. We then describe several biological plausible evidences that support for the uses of frame in wavelet representations. In Subsection 3.3.3, the main ideas of shift-variance wavelet transforms are described. We then modify the existing overcomplete wavelet representations using periodic boundary handling. These representations may not produce enough redundancy when the signal is decomposed into octave subbands. However, we show in the next section that these representations are very useful when the signal is decomposed by using full wavelet decomposing scheme.

3.3.1 Frame

Many problems in signal processing, communications, and information theory deal with linear signal expansions, also known as *subspace approaches*. The corresponding basis functions usually constitute a nonredundant set. It is well known that the use of redundancy in engineering systems improves robustness and numerical stability. The use of redundant linear signal expansions has found widespread use in many different engineering disciplines. Recent examples include sampling theory, A/D conversion, oversampled filter banks, multiple description source coding, error correcting codes, wavelet– and frame– based denoising, quantum detection and estimation, and space–time coding for wireless communications. Frame is generalizations of bases that leads to redundant signal expansions [56]. A *tight frame* is a special case of a frame for which the reconstruction formula is particular simple, and is reminiscent of an orthogonal basis expansion, even though the frame vectors in the expansion are linearly dependent. Specifically, a frame is associated with "oversampling", "overcomplete", or " redundancy." Hence, the use of frames and tight frames rather than bases and orthogonal bases means a certain amount of redundancy exists. In this paper, the concept of frame plays a major role on applying SMs to our coverage optimization method.

3.3.2 Biological Plausible Motivations

Our motivation on the use of frame expansions in multiple classifier approaches is to ensure that model used in our classification system should resemble models that are evidently found in some biological organizations, i.e. biological vision organizations and auditory models.

As argued in biological vision [61] that both retina and cortical data indicate extensive oversampling (manifested by overlap of adjacent receptive fields), especially in the position dependence Gabor representation scheme. Furthermore, results from vision research indicate that an image is represented in logarithmic scale along the frequency axis, so called "Gaborian Pyramid", which is also a basic type of wavelet representations. In auditory systems, it is argued that several models of the early processing in the mammalian auditory organizations are developed under wavelet representations [62, 63]. In particular, a redundant wavelet framework, called *irregular sampling* [63], was addressed for modeling one of the auditory systems.

At the best of our knowledge, we only found one analogy idea of using redundant representations in machine learning [64]. In particular, it was pursued in the context of rule based classification systems. Thus, exploiting redundant multiresolution representations may leads us to new, efficient and biological plausible machine learning algorithms.

3.3.3 Multiple Description Coding with Shift-variant Discrete Wavelet Transforms

Redundant discrete wavelet transform (RDWT) exploits the shift variance property of discrete wavelet packet transform to add desired amounts of redundancy to the original signal or image. This idea has been explored before for image compression application using multiple description coding with redundant discrete wavelet transform [65]. We are now going to summarize the idea and our modifications. The basic concept underlying the construction of a redundant discrete wavelet transform is to expand the number of coefficients in a discrete wavelet transform. There are many possible ways of building a redundant discrete wavelet transform. It can be built by either concatenating several critically subsampled discrete wavelet transforms with different wavelet filters or using an oversampled transform with a fixed set of wavelet filters. It can also be constructed by computing in *parallel* several critically subsampled transforms, each for a different shift of the input signal. In this work, we use the third approach to construct a redundant discrete wavelet transform and an RDWPT.

In discrete wavelet transform, it is known that the convolution–subsampling operation used for perfect reconstruction is considered to be the filtering operation performed on every sample of an input signal at different time shifts. This way, in the traditional discrete wavelet transform, the convolution-subsampling operation with a lowpass filter corresponds to the lowpass filtering at every odd sample (1st, 3rd, 5th, . . .) of the input signal, while the convolution-subsampling operation with a highpass filter corresponds to the highpass filtering at every even sample (2nd, 4th, 6th, . . .). Alternatively, we can implement a new discrete wavelet transform by lowpass filtering at every even sample (2nd, 4th, 6th, . . .) and highpass filtering at every odd sample (1st, 3rd, 5th, . . .), and we will refer to this transform as the *first alternate* wavelet transform. Indeed, it is easy to verify that a redundant representation of the input signal with a redundancy ratio of two can be obtained by combining the wavelet coefficients from these two transforms.

Also, by just swapping the delays of the first alternate wavelet transform, we can construct another discrete wavelet transform, in which we can refer to it as the *second alternate* wavelet transform. Equivalently, the first alternate wavelet transform corresponds to one level decomposition of the traditional discrete wavelet transform on an input shifted left by one sample, if we additionally shifted the output (decomposition coefficients) of the highpass filtered signal right by one sample (see [65, pages 33–36] for further detail). On the other hand, the second alternate wavelet transform corresponds to one level decomposition of the traditional discrete wavelet transform on an input shifted left by one sample, if we additionally shifted the output of the lowpass filtered signal right by one sample. In other words, in the second alternate transform, we keep the even samples (Nth, 2nd, 4th, 6th, . . .) from the lowpass, and odd samples (3rd, 5th, . . . , ($N-1$)th, 1st) from the highpass filtered signal. Apparently, in the case of one level decomposition, the lowpass filtered signals obtained from both the first and second alternate transforms contain all of

the filtered coefficients (2nd, 4th, 6th, . . . , Nth), only these coefficients are not organized sequentially. This occurs to the highpass filtered signal as well. Of course, combining all the wavelet coefficients of the first and second alternate transforms creates a representation with a redundancy ratio of one (or *no redundancy* at all). However, except for the position at the first decomposition level, there are *distinct* coefficients at same position in the trees generated by all the above transforms (the traditional and the first two alternate transforms). As in an example of the second level decomposition, we keep the samples (1st, 5th, 9th, . . .) from the lowpass of the lowpass filtered signal for the traditional transform, the samples (4th, 8th, 12th, . . .) for the first alternate transform, and (($N-2$)th, 2nd, 6th, . . .) for the second alternate transform, respectively. There are distinct coefficients at other subbands as well. Thus, a redundant representation of the input signal with a redundancy ratio of three can be obtained by combining the wavelet coefficients from the above transforms with more than one decomposition level. The above procedure can be extended to two dimensions as well. The only shifts of the 2D signal that are of interest are (column, row) = (0,0),(0,1),(1,0), and (1,1).

Originally, Figure 3.1 (a) to (c) are the traditional and the first two alternate discrete wavelet transforms. These three redundant transforms are first studied in [65]. Despite the transforms given above, we modify the traditional DWT by employing another *periodic boundary handling* method to build a redundant set of wavelet coefficients by circularly left-shifting the signal by one sample and then performing the traditional discrete wavelet transform. We refer to this transform as the *third alternate* wavelet transform. It is easy to verify that a redundant representation of the input signal with a redundancy ratio of 1.5 can be obtained, if we combine the wavelet coefficients from the third alternate transform with any one of the other alternate transforms (the first and second alternate transforms). We obtain the redundant transform with a redundancy ratio of two, if we combine the wavelet coefficients from the third alternate transform with the traditional transform. In this paper, the third alternate wavelet transform (Figure 3.1 (d)) together with the traditional and the first two alternate discrete wavelet transforms will be recursively used up to three level decomposition in order to evaluate the performance of our proposed method. If we organize all wavelet coefficients created from R different transforms, we can obtain a redundant discrete wavelet transform with a redundancy ratio up to R.

3.4 Local Discriminant Frame Expansions

In this section, we first describe the *redundant discrete wavelet packet transform* (RDWPT) in Subsection 3.4.1. It is used in our adaptation of the local discriminant basis (LDB) algorithm to the local discriminant frame expansions (LDFE). In Subsection 3.4.2, the main ideas of our proposed approaches to improve the robustness of LDB are discussed and the *local discriminant frame expansion* algorithm is described.

3.4.1 Redundant Discrete Wavelet Packet Transform

By generalizing discrete wavelet transform to discrete wavelet packet transform, an extension of the redundant discrete wavelet transform to the RDWPT is just straightforward. Note that the discrete wavelet packet transform computed on the input signal without a shift is referred to as the *first transform*. The discrete wavelet packet transform with different circular shifts will be considered as the other transforms (the *redundant data*). As for the extension to local discriminant basis (LDB), we apply the local discriminant basis selection algorithm [55] to each transform of the RDWPT. For the purpose of illustration, we simply demonstrate a one–dimensional RDWPT and its simulated local discriminant bases for four decomposition levels in Figure 3.2. Let us assume that Transform 1 to 4 is subsequently given by the traditional and three alternate transforms, respectively. It can be verified by simple programming that the redundant transform of Figure 3.2 can be obtained with a redundancy ratio of 3.75, if we combine the wavelet coefficients from all the above transforms. Thus, the design of shifting of input data presented for the transforms in Figure 3.1 together with the exploitation of the overcomplete property of wavelet packet transform provides us an efficient method to produce the redundant transform and its multiple descriptions.

3.4.2 Adapting LDB to Local Discriminant Frame Expansions

In our proposed scheme, data is built into several descriptions which are then trained or classified separately. The essential idea underlying the use of the interleaving approach to improve the robustness of multiple classification with a set of local discriminant basis (LDB) coefficients is that the classifiers constructed from the generated descriptions should have an equal and independent probability of classification error. When the classifiers are equally important and independent, the classifier outputs can be combined using majority (unweighted) voting to obtain the final classification output. Previous work [66] showed that the unweighted voting scheme is generally more resilient to overtraining than the weighted method, since overtraining can also be caused from the optimizing process used in the weighted combining scheme. Practically, there is a possibility that our chosen classification algorithm may be too sensitive to features that are derived from one transform more than the other transforms, leading to the need of optimizing the combining weights of the classifier outputs. This problem can be alleviated by dividing a single feature set in any one description into multiple feature subsets which were each derived from a different transform. In particular, each feature subset corresponds to a different part in the signal. Thus, it seems to be necessary to use many transforms (to derive multiple feature subsets) in one description, instead of using only one transform. As discussed in [44], making descriptions individually good and yet independent of each other is the key idea of the multiple description method. In this paper, we build equally good classifiers from a set of powerful distance based classifiers using highly informative (equally important) descriptions as input feature sets. In

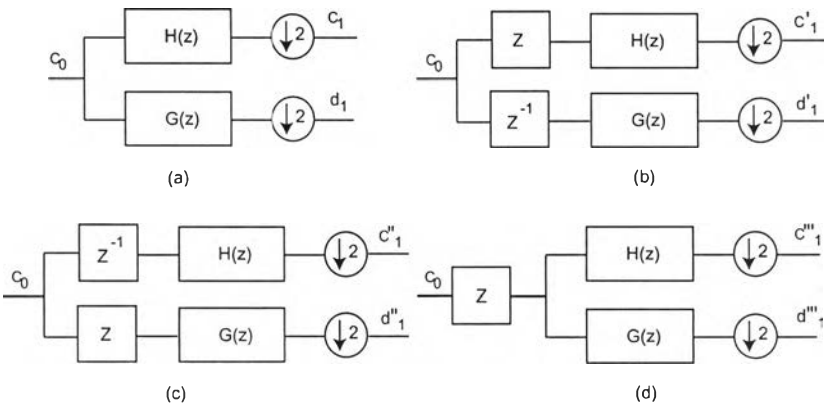

Figure 3.1: Four methods of filtering and subsampling for one level decomposition discrete biorthogonal wavelet transform. a) Traditional wavelet transform. b) First alternate wavelet transform. c) Second alternate wavelet transform. d) Third alternate (left-shifting) wavelet transform. c_0 is a set of wavelet coefficients obtained from the previous level. c_1 and d_1 are the lowpass and highpass filtered signals subsampled by a factor of two. z and z^{-1} are the left and right circular shifting operators, respectively.

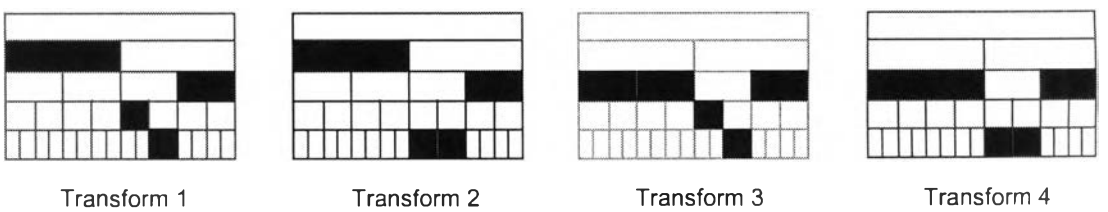

Figure 3.2: Results of four decomposition levels of a redundant wavelet packet transform and its simulated LDB functions.

order to build independent classifiers from the descriptions of roughly equal importance (by the use of many transforms), the dependency (correlation) between descriptions has to be eliminated by interleaving the subset of LDB (MDB) coefficients obtained from different transforms. It is widely accepted in channel coding that interleaving redundancy data is one of the successful methods to combat channel error, e.g., turbo coding. To apply this to image analysis, spatially dispersed 2D LDB coefficients from different transforms are grouped together, selected the most discriminant basis functions, and used as one description.

In particular, we try to invent descriptions suitable for building classifiers such that good diversities among classifiers occur as much as possible. In this work, we first group transforms so that a set of descriptions can be built from such transforms. More precisely, the transforms used at each spatial region of all the descriptions must be different, one transform is at a discriminant level higher than the rest of the transforms in the subset, and the transforms should have different discriminant levels. After building the first description in the subset, we build other descriptions by permuting the transforms of the spatial regions used in the preceding descriptions. We may continue to build more descriptions using a different subset of transforms. As shown in Figure 3.3, four discrete wavelet packet transforms and their discriminant levels are presented.

As illustrated in Figure 3.4, based on the information presented in Figure 3.3, the first description is constructed by grouping four different transforms, the transforms in any descriptions have different discriminant levels, and the first transform has its maximum classification strength specialized at the spatial region 1. For the next descriptions, we assign the first transform to be specialized at other spatial regions, and the process is repeated for other transforms with lower classification strength as well. When all the transforms within the transform subset are applied, the set of descriptions is obtained, as shown in Figure 3.4. Here, discriminant level 1 means the highest discriminant power, and we consider the original transform (the *first* transform) having the highest discriminant power. In this work we expect that the power of discriminant is specified by the number of the MDB functions. Thus, we allocate the highest number of the MDB functions for the spatial regions with discriminant level 1, the next highest number of the MDB functions to the next discriminant level, and so on.

The reason underlying the assignment of the discriminant level is that the more important information about each spatial region should be contained in a larger number of descriptions than the less important information. In other words, we should add unequal amounts of redundancy (protection) to different elements of the input features (or equivalently the MDB functions). By assigning the discriminant level, the more important LDB coefficients are protected with a stronger redundancy transform, and the less important LDB coefficients are protected with a weaker redundant transform. As in the example above, the most important data of spatial region 1 is likely to be protected by the redundant transform consisted of the three transforms. The next important data is likely to be protected by the redundant transforms consisting Transform 1 and 2. With no protection, the less important information

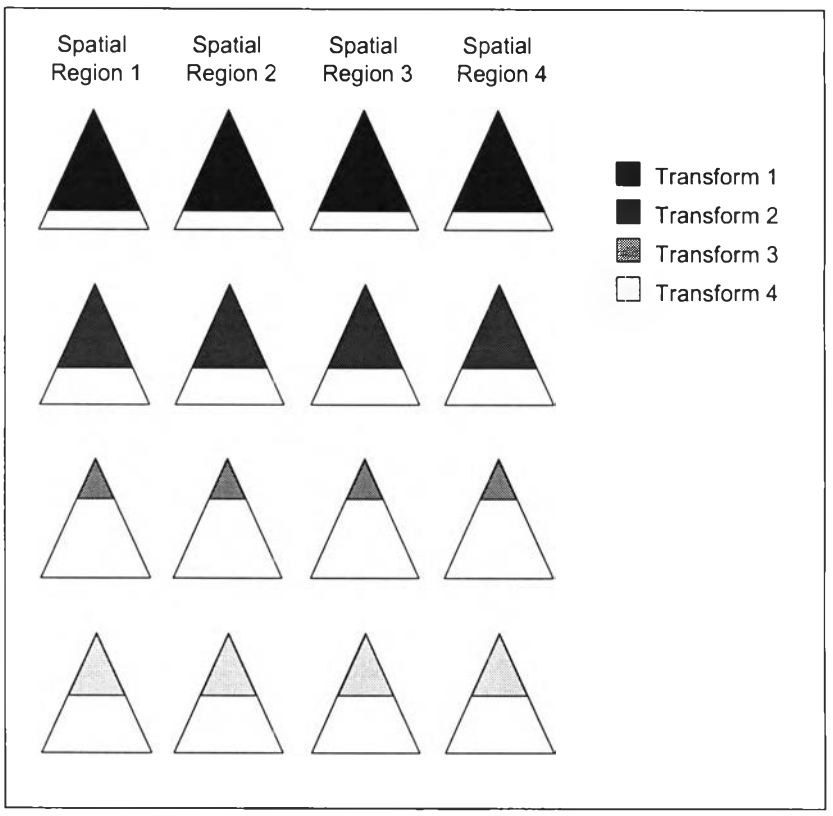

Figure 3.3: Example of four transforms and their spatial discriminant levels. The gray area indicates the size of the most discriminant basis functions (or equivalently its power of discriminant).

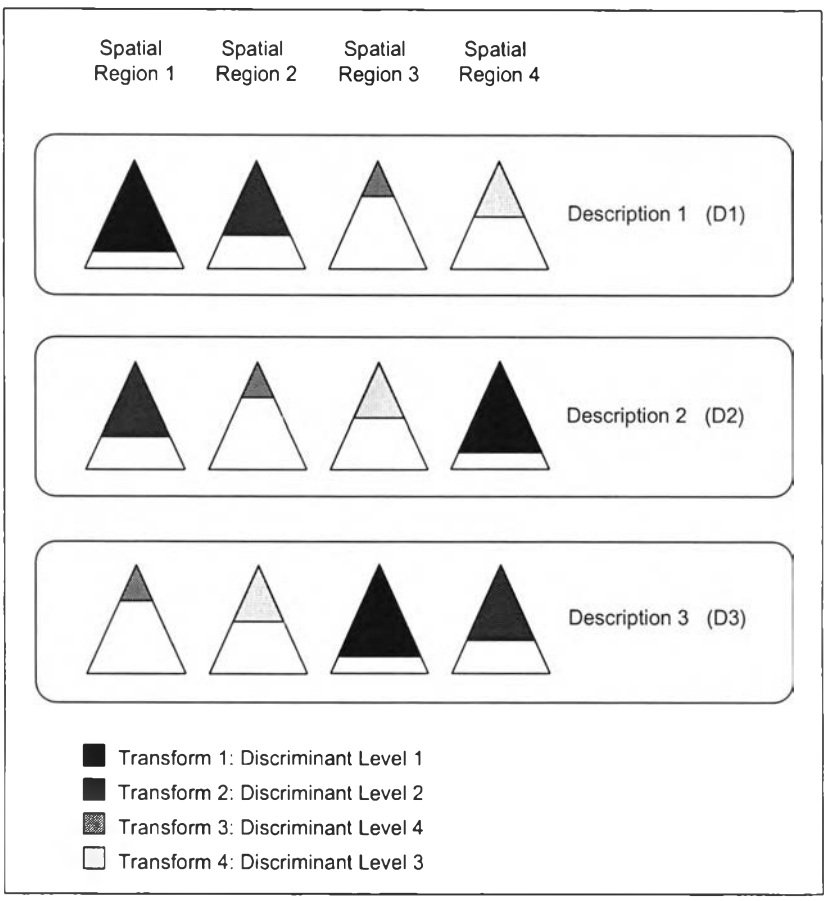

Figure 3.4: Example of three spatially dispersed descriptions extracted by using local discriminant frame expansions of four transforms.

is represented by the set of less important LDB coefficients obtained from Transform 1. The least important information or the useless part of the data (the least important LDB coefficients) is usually discarded beforehand by the LDB algorithm. Interleaving each description to have its maximum classification strength specialized at the different spatial region, we can independently build a set of diverse classifiers using descriptions as input feature sets. This way, features are protected in an unequal loss protection manner, and this also applies to the concept representations of the class information as well. Through the discriminant level setting, in this work the total number of the MDB functions of all descriptions is intentionally set to be equal. Keeping the total number of the interleaving MDB functions to be equal, independent descriptions of roughly equal discriminant are likely to be obtained. Thus, the discriminant level assignment becomes necessary, if we want to further reduce the dimensionality of the problem.

3.5 Multiple Description Coding for Multiple Classifier Systems with Local Discriminant Frame Expansions

So far, we have explained how RDWPT can be used in our adaptation of LDB algorithm to *local discriminant frame expansion* (LDFE) algorithm. We now summarize the algorithm in Figure 3.5. It should be noted that Step 0 to 4 are concerned with the training process, and Step 5 is the recognition process.

Related to the communication model for classification, our class information is less distorted to the errors caused by input features because LDB itself selects the optimal coordinates suitable for using as input features, and this property is extended to LDFE as well. It should be noted that, in LDFE algorithm, each classifier has invariant classification for points that are different from the training points only in the unselected dimensions due to the shift-variant property of wavelet transforms. This way, each classifier generalizes its classification in a different way. This is similar to the random subspace method (see [25] for details). As a result, the interesting properties on equally important and independent projectability, minimum enrichment (according to the MDB functions selection method and the Mahalanobis classification), and guaranteed uniformity make LDFE a good candidate method to use for constructing efficient multiple classifier systems.

3.6 Experimental Results

In this section, the performance of our proposed scheme and other classifier systems on the MSTAR public release data set are compared. The MSTAR public release data set contains high resolution synthetic aperture radar data collected by the DARPA/Wright laboratory Moving and Stationary Target Acquisition and Recognition (MSTAR) program. The data set contains SAR images of three different types of military vehicles – BMP2 armored personal carriers (APCs), BTR70 APCs, and T72 tanks. The samples of SAR images

at different orientations are shown in Figure 3.6. The size of the images is originally at 128x128.

Tables 3.1 and 3.2 detail the training and testing sets, where the depression angle means the look angle pointed at the target by the antenna beam at the side of the aircraft. Based on the different depression angles SAR images acquired at different times, the testing set can be used as a representative sample set of the SAR images of the targets for testing the recognition performance.

The problem of automatic target recognition (ATR) is a difficult one. Research and development in the past decade have proposed several solutions to ATR. For example, the feature extraction method based on modified differential box-counting (MDBC) [67] was proposed for estimating the fractal dimension of the original images. These MDBC features were then used to train and test feed-forward neural networks (NN). Additionally, Bayesian classification algorithm based upon a family of conditionally Gaussian signal models (CGSM) for SAR imagery [68] were used to jointly estimate both target type and target pose. Two modified hidden Markov model (HMM) methods [69] were also proposed for solving this problem. Their best percent recognition accuracy is 96.76 (see other proposed methods and their performance comparisons in [69]). Moreover, perceptron, optimal hyperplane, and support vector machines (SVM) were the three strategies of learning methods and representations proposed for SAR ATR [70]. Their classification performances were reported for 80 x 80 target window with percent recognition accuracy 88.06, 90.55, and 90.99, respectively. In their classifier system, the SVM with Gaussian kernel was trained by the Adatron learning algorithm and constructed as *one class in one network*. For the purpose of evaluation completeness, these available recognition accuracy were summarized from References [67–70] in order to be used for comparisons with our next empirical experiments.

Empirically, we studied the performance of ECOC using the original data and the most important LDB coefficients extracted from the original data. The codes used in this experiment setting were constructed using the exhaustive technique [4]. The base classifier used to learn each bit of the codeword was the support vector machine. These SVM for ECOC (ECOC–SVM) experiments were implemented by the software developed by Schwaighofer and available from [71]. software Note that the SVM with linear kernel was used in this evaluation.

We also studied the performance of Adaboost with stochastic weighting [35, 72, 73] using both the original data and the most important LDB coefficients. In our Adaboost experiments, the number of weak models implemented in our experiments was 30. The weak model was one hidden-layer backpropagation network, each was trained for 500 iterations using various numbers of hidden nodes (5, 6, and 7 hidden nodes). We ran each Adaboost experiment for six times and then averaged for the recognition accuracy results.

To demonstrate the capability of the local discriminant basis (LDB) and local discriminant frame expansion (LDFE) schemes, we further conducted two series of the experiments

Algorithm:

Step 0: Partition input pattern to $\mathcal{L}$ spatial nonoverlapped regions or subimages x_l, $l = 1, ... \mathcal{L}$.

Step 1: Expand x_l using basis from the r^{th} transform (redundant discrete wavelet packet transform), $r = 1, ..., R$.

Step 2: Compute the most discriminant basis (MDB) functions $\{v_n\}^{lr}$, $n = 1, \ldots, N^{lr}$, where N^{lr} is the number of the MDB functions predefined by a discriminant level.

Step 3: Build spatially dispersed descriptions Υ_k from $\{v_n\}^{lr}$, $k = 1, \ldots, K$.

Step 4: Use Υ_k to construct the k^{th} nearest Mahalanobis distance classifier.

Step 5: Use majority vote to make final decision for K classifiers.

Figure 3.5: Multiple Description Pattern Analysis using Local Discriminant Frame Expansion.

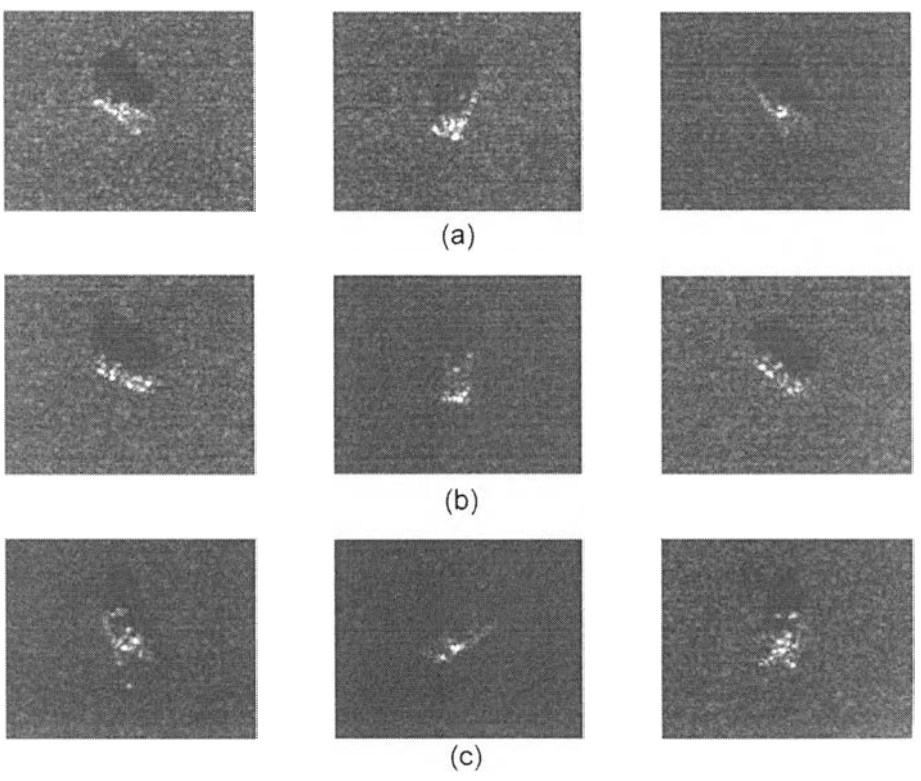

Figure 3.6: Sample SAR images of military vehicles. (a) BMP2 APCs, (b) BTR70 APCs, and (c) T72 tanks.

Table 3.1 MSTAR images comprising training set.

	Vehicle No.	Serial No.	Depression	Images
BMP-2	1	9563	17^0	233
	2	9566		231
	3	c21		233
BTR70	1	c71	17^0	233
T-72	1	132	17^0	232
	2	812		231
	3	s7		228

Table 3.2 MSTAR images comprising testing set.

	Vehicle No.	Serial No.	Depression	Images
BMP-2	1	9563	15^0	195
	2	9566		196
	3	c21		196
BTR70	1	c71	15^0	196
T-72	1	132	15^0	196
	2	812		195
	3	s7		191

using classifiers with high discriminant power. In the LDB scheme, we employed Mahalanobis distance to classify input features obtained from selecting the most important LDB coefficients. The purpose of including this classifier as the baseline for comparison was to investigate the performance of our LDFE scheme over the original LDB scheme. Note that throughout the experiments with LDB and LDFE methods, we used the first order Coiflet filters (comparable to 6-tab Daubechies filters with respect to the support length) for computing 2D discrete wavelet packet transform into three decomposition levels.

In the LDFE scheme, we first created 10 redundant versions of the 2D wavelet packet decomposition functions for each of the spatial regions or subimages (in these experiments, we set the number of subimages equal to 4). Then, we computed LDB for each transform version of each subimage using the discriminant measure (3.3). After the LDB selection, each LDB coordinate was sorted in terms of its discriminant power using the relative entropy (3.4). Next, spatially dispersed descriptions were built from the 10 versions of the top LDB coordinates. In this evaluation, we used the first four transforms (Transform 1 to Transform 4) as the first transform subset, Transform 4 to Transform 7 as the second transform subset, and Transform 7 to Transform 10 as the third transform subset. The first three descriptions (D1-D3) were constructed from the first transform subset, while the next three descriptions (D4-D6) were constructed from the second transform subset, and so on. In particular, each transform in each transform subset was sequentially assigned a discriminant level (or equivalently the number of the MDB functions). We set the number of the MDB functions used by each transform to be slightly different. For example, in the case of the first transform subset, Transform 1 had been assigned the highest discriminant level (discriminant level 1), the discriminant levels of Transform 2 and Transform 4 were equal and at the discriminant level 2, and Transform 3 had the lowest discriminant level. The difference of the number of the MDB functions between each discriminant level was set to be equal to two. After all of the MDB functions corresponding to their subimage locations were selected according to their predefined transforms and discriminant levels, we grouped all of the MDB functions into one description. Each description was built by permuting the transforms over all of the subimage locations (see Figure 3.3 and Figure 3.4 for the permutations of the transforms used in our experiments). This was applied to the second and the third transform subsets as well to construct the other descriptions. This way, we could obtain 7 spatially dispersed descriptions, and used them as input features for building 7 Mahalanobis distance based classifiers. Finally, we used majority vote to combine the output decisions of the classifiers.

Note that filtering and subsampling methods shown in Figure 3.1 were used for building our proposed 2D redundant discrete wavelet packet transform. We now summarize the detail of our proposed redundant transform illustrated in Figure 3.7. Transform 1 is the discrete wavelet transform without a shift. Transform 2 to Transform 4 (Figure 3.7 (a) to (c)) perform the left-shifting discrete wavelet decomposition either at the horizontal or the vertical direction. Similarly, Transform 5 to Transform 7 (Figure 3.7 (d) to (f)) perform the second alternate wavelet transform, and Transform 8 to Transform 10 (Figure 3.7 (g) to (i))

perform the first alternate wavelet transform either at the horizontal or the vertical direction. Note that z_1 and z_2 are the one unit horizontal and vertical circular left-shifting, respectively.

Four different image sizes from 32x32 to 80x80 had been evaluated in our experiments. Image chips of different image sizes were constructed by extracting a small rectangular region at the center of the MSTAR images. The reason for windowing target images is that the classification should be sensitive to a region corresponding to the vehicle, not to background clutter or target shadow. The larger the image chip is used, the more target, shadow, and background clutter pixels are included. Inversely, portions of shadow and background are eliminated and the target occupies a larger portion of image chip, when a smaller image chip is used. Tables 3.3 and 3.4 detail the recognition performance through confusion matrices for 80 x 80 images of the CGSM and our proposed scheme, respectively. As presented in Table 3.5, our proposed scheme gives the best overall performance among all approaches at almost all image sizes. These results are based on the best performance of each classifier system evaluated at either the optimal number of hidden nodes (Adaboost) or the optimal number of the LDB coefficients. As we expected, the performance of our proposed scheme is better than the performance of ECOC−SVM and the SVM classifier system in [70]. As addressed in [4], unless the number of classes is at least five, it is too difficult to maximize the Hamming distance between codewords and at the same time minimize the dependence between the errors of the individual binary classifier units. Here, we presented an extreme case for ECOC−SVM, in which any single binary error could lead to an ambiguity in a codeword such that ECOC−SVM would not be able to identify the target class. In our future work, we will incorporate LDFE with the ECOC−SVM method, and perform a direct comparison of ECOC−SVM with our method for a SAR ATR problem with a larger number of classes. Moreover, the performance of our proposed method converges faster than Adaboost because a very large number of models is needed for Adaboost learning/classifying to be converged. From the experimental results, we can see that the LDB method is an efficient technique for improving accuracy for both ECOC−SVM and Adaboost methods. In particular, the LDB method is an effective feature selection technique that can prevent ECOC−SVM from overtraining. Noise is one of the main shortcomings in Adaboost (see [27] and the references therein). Since LDB method is considered to be one of the noise reduction methods, the Adaboost accuracy were improved by employing the LDB method to select features. It should be noted that, at the 32x32 image size, the Mahalanobis classifier was too sensitive to the MDB functions. This is due to the fact that adding just a few irrelevant features could drastically change the outputs of the distance based classifier, and also reduce its accuracy (see [13] and the references therein). In case of the 32x32 image size, the features we used consisted of too many irrelevant features (the number of the features was about one−third of the input dimensions). At the larger image size, the performance of our proposed method outperformed other methods, since all the relevant features could be obtained from a larger collection of the LDB coefficients. Additionally, the performance

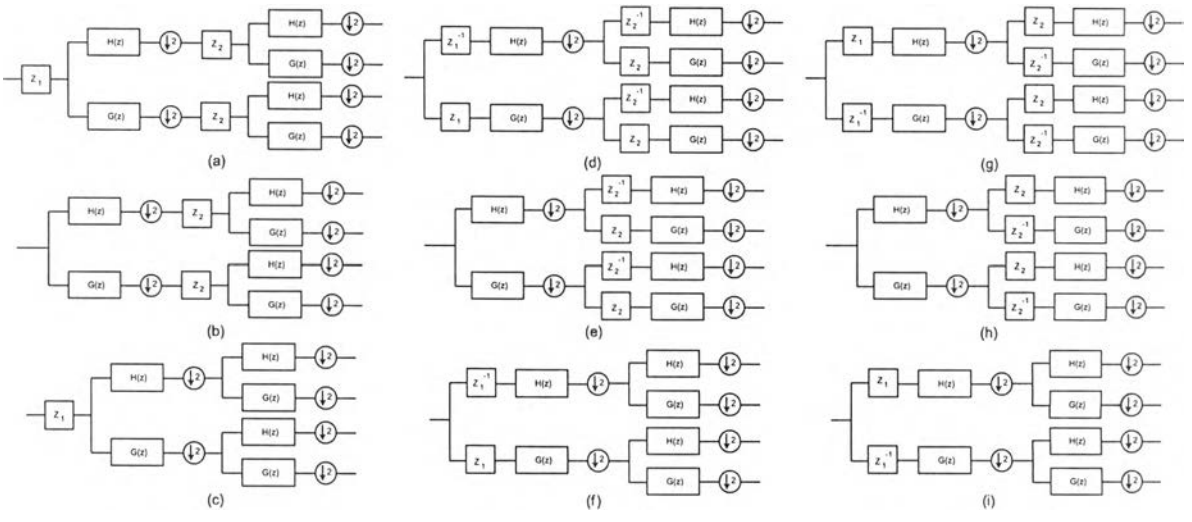

Figure 3.7: One level decomposition of redundant versions of 2-D discrete wavelet transforms. a) to i) Transform 2 to Transform 10. z_1 and z_2 are the one unit horizontal and vertical circular left-shifting, respectively.

Table 3.3: The CGSM method : Recognition test of a three class problem for 80 x 80 images.

	BMP-2	BTR-70	T-72	Percent
BMP-2	580	0	7	98.81
BTR-70	12	183	1	94.90
T-72	0	0	582	100
Avg. -	-	-	-	98.53

Table 3.4: Multiple description pattern analysis using local discriminant frame expansions with 7 descriptions: Recognition test of a three class problem for 80 x 80 images.

	BMP-2	BTR-70	T-72	Percent
BMP-2	584	2	1	99.49
BTR-70	1	194	1	98.98
T-72	1	1	580	99.66
Avg. -	-	-	-	99.49

results in term of percent of correct recognition as functions of number of descriptions (or equivalently the redundancy ratio) and image size are shown in Figure 3.8. From the experimental results, ambiguities occurred less often on every odd number of descriptions.

When using the LDFE method, one important parameter to be determined is how many MDB functions should be selected in each description in order to guarantee uniformity. In order to verify the effect of different numbers of MDB functions, we ran the LDFE experiments by varying the numbers of MDB functions from 300 to 380. As shown in Figure 3.9, the results show that the use of too many MDB functions leads to overtraining, especially with the small image sizes, e.g., 32 x 32 and 48 x 48. For larger image sizes, e.g., 64 x 64 and 80 x 80, the biased classification occurs when we select too few MDB functions, but the overtraining problems still exist if we select too many MDB functions. Therefore, the optimization of the size of the MDB functions for a given classification problem should be further studied.

3.7 Numerical Comparison

As shown in Figure 3.10, let's consider some quantities involved with the computational evaluation of the MCS methods. For support vector classifiers, the calculation of the Lagrange multiplier parameters has the complexity $\mathcal{O}(p^3)$ for its computation by direct method and $\mathcal{O}(p^2)$ by iterative method. Note that this is the training complexity of SVM. The evaluation (testing) complexity of SVM consists of a kernel calculation and its comparison of the new sample and all support vectors p^s for that classifiers. A kernel has complexity about $5.5d$, leading to the complexity of kernel calculation for all support vectors $p^s(5.5d)$. Moreover, there is an additional calculation for multi-class comparison with the computational complexity $c + 2cp^{s'}$, where $p^{s'} = \sum_{j=1}^{c} p^j$, and p^j is the number of support vectors for the classifier describing class j. For the sake of simplicity, we simply the complexity of SVM to $\mathcal{O}(dp)$, since $p^{s'} < p$.

For feedforward neural networks, the complexity is dependent of p and number of iteration T while training, but it is independent of p and T while testing. The computational complexity [74] is $2q + 3.5h$, where q and h are the connections between units and bias terms, respectively. We can simplify the complexity of neural networks to $hdpT$ and hd for training and testing respectively, where $q = hd$. For multiple system, e.g., Adaboost, we increase the computational complexity of neural network algorithm by K'-fold, where K' is the number of classifiers constructed according to Adaboost algorithm.

Next, we evaluate the computational complexity of Mahalanobis distance based classifier since it is mainly used in our experiments. The complexity of this type of classifiers depends on distance evaluation. Here, the complexity of a distance calculation $D(z, x)$, where z, x are d-dimensional vector, is approximated $3d$ (note that most of the complexity evaluations here are taken from [13, 74–76]). For k-NN classifiers, the complexity of distance calculation to all prototypes is thus $3d$, and the complexity of storing the minimum

Table 3.5: Comparison of difference methods in overall percentage of images correctly recognized as a function of image size.

Methods / Image Size	32x32	48x48	64x64	80x80
MDBC+NN [67]	75.88	N/A	N/A	N/A
CGSM [68]	N/A	N/A	N/A	98.53
ECOC−SVM(original)	84.46	90.16	91.76	92.70
ECOC−SVM(MDB)	85.42	90.81	92.51	92.97
Adaboost(original)	88.24	93.35	93.48	93.68
Adaboost(MDB)	89.66	93.16	94.26	93.97
Mahalanobis Dist.(MDB)	73.78	95.24	97.14	97.29
our proposed scheme Mahalanobis Dist.(LDFE)	84.69	98.53	99.34	99.49

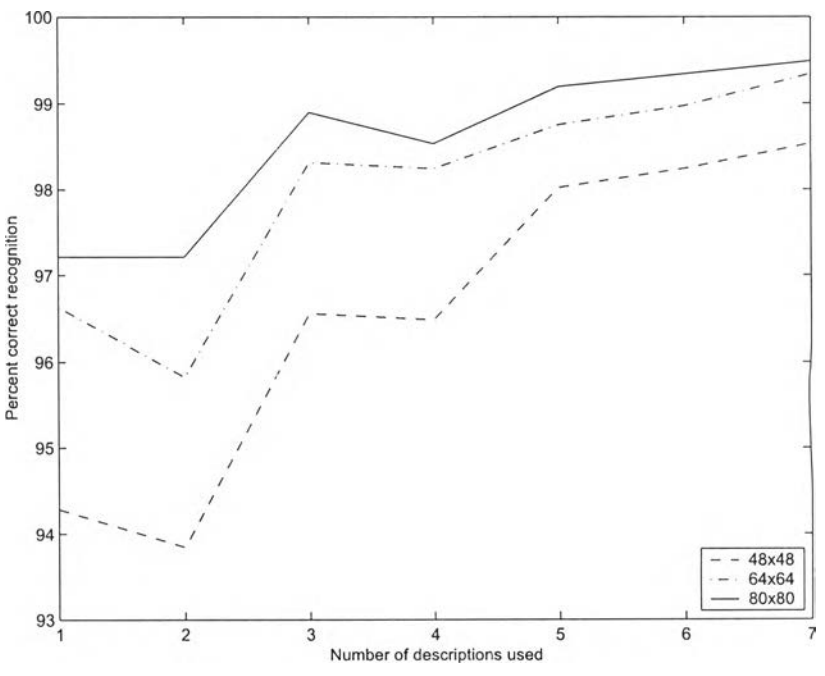

Figure 3.8: The performance of multiple description pattern analysis using local discriminant frame expansions at various image sizes.

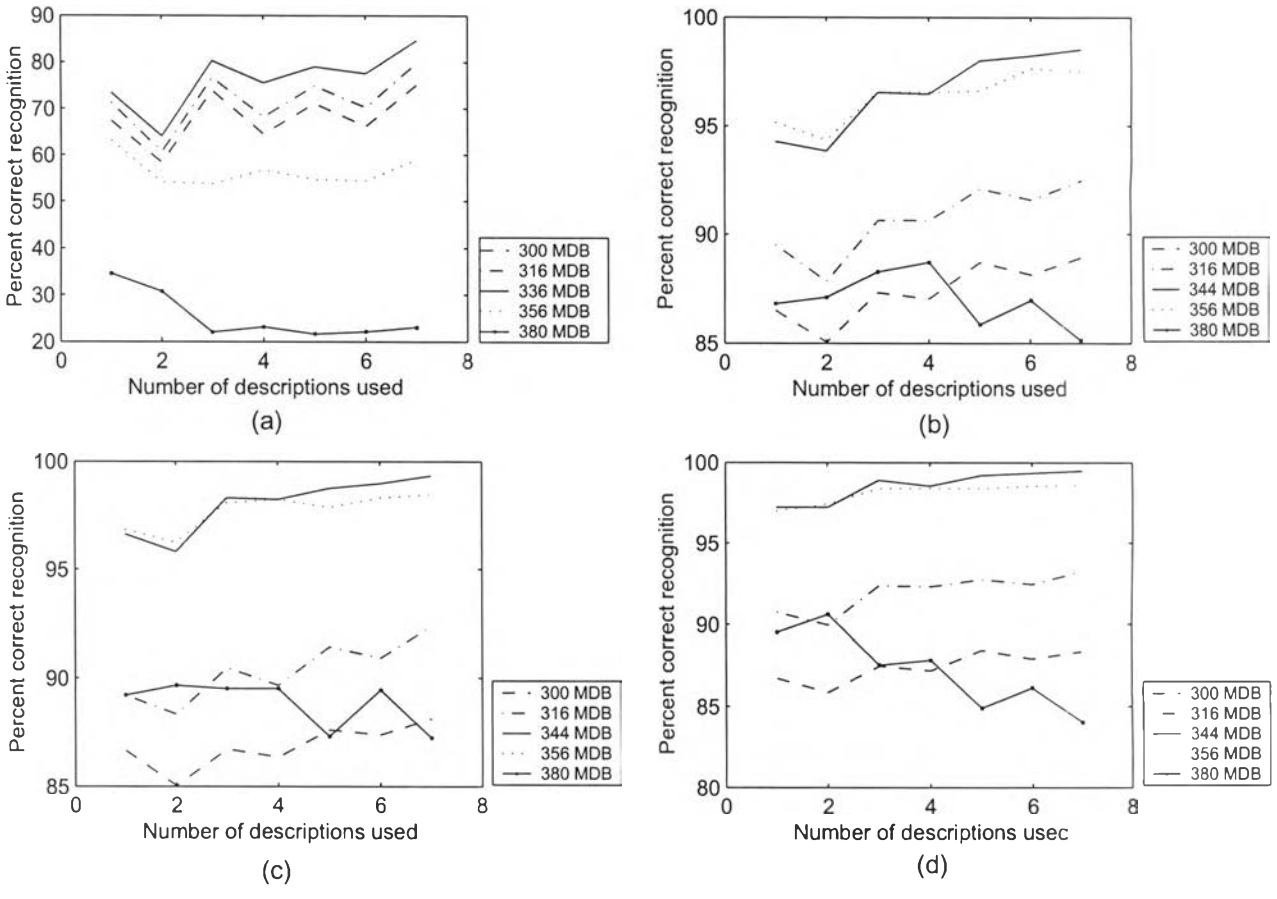

Figure 3.9: The performance of multiple description pattern analysis using local discriminant frame expansions at different numbers of the most discriminant basis functions. a) 32 x 32 image size. b) 48 x 48 image size. c) 64 x 64 image size. d) 80 x 80 image size.

p	The number of training samples in the training set
d	The number of features (dimensions) in each sample
n, M	The number of MSB dimensions in each sample
N	The number of original dimensions in each sample
c	The number of classes to be distinguished
K	The number of classifiers in our proposed method
K'	The number of classifiers for Adaboost
h	The number of hidden nodes of neural networks
T	The number of iterations of neural network training

Figure 3.10 Notation for parameters used in complexity evaluation of MCS methods.

in a sorted list of k nearest prototypes is $p \log_2 k$. The total complexity therefore is $p3d$ when $k = 1$. Similarly, we simplify the complexity of algorithm by measuring them in order according to the Landau symbol $\mathcal{O}$, e.g., the complexity of k-NN classifiers can be replaced by $\mathcal{O}(np)$, when we used only n local discriminant features with $n < d$.

It should be noted that the above complexity discussion is for the two-class case. For multi-class case, we increase the computational complexity of some algorithms by c-fold. Now we summary the computational complexity of the evaluated MCS methods for training and testing procedures in the Tables 3.6 and 3.7, respectively.

Now, we are ready to evaluate the computational complexity of the MCS methods for SAR ATR. In the 3-class SAR ATR problem, $p = 1621$, $n \cong 320$, N = 1024, 2304, 4096, and 6400, $K = 8$, $K' = 30$, $h = 7$, $T = 800$. It is easily to verify that $\mathcal{O}(p^3 c) > \mathcal{O}(Np)$. In the same way, we detail the approximate computational complexities of the other methods in Table 3.8.

Therefore, we can arrange the algorithms from high to low computational complexity as follows:

> Adaboost (original) > Adaboost (MDB) > ECOC-SVM (MDB) > ECOC-SVM (original) > Mahalanobis Classifier (LDFE) > Mahalanobis Classifier (MDB).

We finally arrange the degree of the computational complexity and its recognition accuracy of each MCS method in Table 3.9.

3.8 Conclusions

In this chapter, we proposed an alternative method to ECOC. Our proposed scheme was inspired from the framework of transmitting data over heterogeneous networks, especially wireless networks. To our knowledge, we are the pioneers in applying MD coding to pattern recognition. Based upon the experimental results, our proposed scheme gave the best performance among the state-of-the-art multiple classifier systems. Since our proposed scheme is deterministic by nature, its computation is thus more competitive than stochastic based multiple classifier systems, e.g., Adaboost or random subspace method.

Table 3.6 Comparison of training computational complexity.

Methods / Image Size	Feature Extraction	Training	Approximate Complexity
ECOC−SVM(original)	–	$\mathcal{O}(p^3c)$	$\mathcal{O}(p^3c)$
ECOC−SVM(MDB)	$\mathcal{O}(npc\log n)$	$\mathcal{O}(p^3c)$	$\mathcal{O}(npc\log n + p^3c)$
Adaboost(original)	–	$\mathcal{O}(hNpcK'T)$	$\mathcal{O}(hNpcK'T)$
Adaboost(MDB)	$\mathcal{O}(npc\log n)$	$\mathcal{O}(hnpcK'T)$	$\mathcal{O}(npc(hK'T + \log n))$
Mahalanobis Dist.(MDB)	$\mathcal{O}(npc\log n)$	–	$\mathcal{O}(npc\log n)$
our proposed scheme Mahalanobis Dist.(LDFE)	$\mathcal{O}(npcK\log n)$	–	$\mathcal{O}(npcK\log n)$

Table 3.7 Comparison of evaluation (testing) computational complexity.

Methods / Image Size	Feature Extraction	Testing	Approximate Complexity
ECOC−SVM(original)	–	$\mathcal{O}(Np)$	$\mathcal{O}(Np)$
ECOC−SVM(MDB)	$\mathcal{O}(nc\log n)$	$\mathcal{O}(np)$	$\mathcal{O}(np(1 + c\log n))$
Adaboost(original)	–	$\mathcal{O}(hNcK')$	$\mathcal{O}(hNcK')$
Adaboost(MDB)	$\mathcal{O}(nc\log n)$	$\mathcal{O}(hncK')$	$\mathcal{O}(nc(hK' + \log n))$
Mahalanobis Dist.(MDB)	$\mathcal{O}(nc\log n)$	$\mathcal{O}(npc)$	$\mathcal{O}(nc(p + \log n))$
our proposed scheme Mahalanobis Dist.(LDFE)	$\mathcal{O}(ncK\log n)$	$\mathcal{O}(npcK)$	$\mathcal{O}(ncK(p + \log n))$

Table 3.8: Comparison of computational complexity for the MCS methods implemented in our experiment.

Methods / Image Size	Training	Testing	Approximate Complexity
ECOC−SVM(original)	$\mathcal{O}(p^3c)$	$\mathcal{O}(Np)$	$\mathcal{O}(p^3c)$
ECOC−SVM(MDB)	$\mathcal{O}(npc\log n + p^3c)$	$\mathcal{O}(np)$	$\approx \mathcal{O}(p^3c)$
Adaboost(original)	$\mathcal{O}(hNpcK'T)$	$\mathcal{O}(hNcK')$	$\mathcal{O}(hNpcK'T)$
Adaboost(MDB)	$\mathcal{O}(npc(hK'T + \log n))$	$\mathcal{O}(hncK')$	$\approx \mathcal{O}(hnpcK'T)$
Mahalanobis Dist.(MDB)	$\mathcal{O}(npc\log n)$	$\mathcal{O}(npc)$	$\approx \mathcal{O}(npc\log n)$
our proposed scheme Mahalanobis Dist.(LDFE)	$\mathcal{O}(npcK\log n)$	$\mathcal{O}(npcK)$	$\approx \mathcal{O}(npcK\log n)$

Table 3.9: Comparison of accuracy (in overall percentage) and computational complexity for the MCS methods implemented in our experiment.

Methods / Image Size	32x32	48x48	64x64	80x80	Complexity
ECOC−SVM(original)	84.46	90.16	91.76	92.70	Medium
ECOC−SVM(MDB)	85.42	90.81	92.51	92.97	Medium
Adaboost(original)	88.24	93.35	93.48	93.68	Very High
Adaboost(MDB)	**89.66**	93.16	94.26	93.97	High
Mahalanobis Dist.(MDB)	73.78	95.24	97.14	97.29	Very Low
our proposed scheme Mahalanobis Dist.(LDFE)	84.69	**98.53**	**99.34**	**99.49**	Low