2007

Efficient Algorithms For Correlation Pattern Recognition

Pradeep Ragothaman
University of Central Florida

EFFICIENT ALGORITHMS FOR CORRELATION PATTERN RECOGNITION

by

PRADEEP RAGOTHAMAN
B.E. University of Pune, 1999
M.S. University of Central Florida, 2003

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2007

Major Professor: Wasfy B. Mikhael

ABSTRACT

The mathematical operation of correlation is a very simple concept, yet has a very rich history of application in a variety of engineering fields. It is essentially nothing but a technique to measure if and to what degree two signals match each other. Since this is a very basic and universal task in a wide variety of fields such as signal processing, communications, computer vision etc., it has been an important tool. The field of pattern recognition often deals with the task of analyzing signals or useful information from signals and classifying them into classes. Very often, these classes are predetermined, and examples (templates) are available for comparison. This task naturally lends itself to the application of correlation as a tool to accomplish this goal. Thus the field of Correlation Pattern Recognition has developed over the past few decades as an important area of research.

From the signal processing point of view, correlation is nothing but a filtering operation. Thus there has been a great deal of work in using concepts from filter theory to develop Correlation Filters for pattern recognition. While considerable work has been to done to develop linear correlation filters over the years, especially in the field of Automatic Target Recognition, a lot of attention has recently been paid to the development of Quadratic Correlation Filters (QCF). QCFs offer the advantages of linear filters while optimizing a bank of these simultaneously to offer much improved performance.

This dissertation develops efficient QCFs that offer significant savings in storage requirements and computational complexity over existing designs. Firstly, an adaptive algorithm is presented

that is able to modify the QCF coefficients as new data is observed. Secondly, a transform domain implementation of the QCF is presented that has the benefits of lower computational complexity and computational requirements while retaining excellent recognition accuracy. Finally, a two dimensional QCF is presented that holds the potential to further save on storage and computations. The techniques are developed based on the recently proposed Rayleigh Quotient Quadratic Correlation Filter (RQQCF) and simulation results are provided on synthetic and real datasets.

To my parents.

ACKNOWLEDGMENTS

There are a number of people who have played important roles in helping me reach this far. Firstly, I would like to thank my advisor, Dr. Wasfy B. Mikhael for his guidance and support for these past many years. He has been most patient, helpful and inspiring in his many roles as teacher, mentor and advisor.

I am grateful to Dr. Robert R. Muise and Dr. Abhijit Mahalanobis for getting me started on this research path and for their feedback and support over the course of this dissertation. I would also like to acknowledge the other wonderful teachers that have helped in my technical growth. I would like to thank Dr. Thomas Yang for his help, and my committee members, Dr. Ram Mohapatra and Dr. Issa Batarseh for their time and their feedback.

I would like to acknowledge the various colleagues and friends that have been a part of my life over the past several years at UCF, especially Venky, Parvez, Anna, and Dimitrios. Pravin, Thomas, Guangyu, Yuan, Moataz, Raghu, Aditya, Emad, Ramy and Jimmy made the lab a fun place to work in. I would also like to acknowledge my friends outside the lab and UCF – there are too many to list individually here. Finally, I would like to thank my family for their love, support and patience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

Pattern recognition refers to the task of examining data for patterns and then classifying these patterns into classes or categories. While human beings, and other animals to a lower degree have a very powerful and robust pattern recognition system inherently built into them, researchers have looked into different ways of building similar systems as a first step to the ultimate goal to endow machines with intelligence and the ability to learn. Thus research into pattern recognition has become a big area under the broader umbrellas of machine learning and artificial intelligence.

Research into pattern recognition has followed three broad approaches – statistical, syntactic and neural. The statistical approach is based on characterizing the underlying systems that generate the patterns in probabilistic terms, and using decision-theoretic techniques. On the other hand, syntactic or structural approaches are based on studying the relationships between features and by using grammatical inference and parsing. Finally, neural approaches try to mimic the human brain by trying to build highly parallel and interconnected systems based on the human brain. The pattern recognition area has matured a lot since its inception and great deal of literature has appeared in the form of books and monographs, [1] – [60].

Correlation pattern recognition has recently emerged as an important sub-area of statistical pattern recognition research [61]. As the name suggests, this area of research deals with methods

that make use of the mathematical operation of correlation to measure the degree of similarity between a stored or created template or reference signal and a new unknown signal. On the other hand, it is not to be interpreted as simple matched filtering. A review of the literature in this field reveals a plethora of techniques that have been derived by using concepts from several fields such as linear algebra, signal processing, estimation theory etc, [62]. Although most of the techniques developed have been in the Automatic Target Recognition (ATR) area, they have started gaining popularity in other areas, especially Biometrics.

The field of ATR has received considerable attention over the years. The process of detecting and classifying objects of interest embedded in background clutter is a challenging task, especially since clutter is often the dominant component of Forward Looking Infrared (FLIR), Synthetic Aperture Radar (SAR) and Laser Radar (LADAR) images. There are many approaches to ATR that have been reported in the literature. Some techniques are based on modeling target signatures usually obtained after segmentation of images to extract objects of interest, [63]-[71]. Others involve feature extraction to implicitly recognize targets, [72]-[74]. In addition, many techniques have been reported that use neural networks, statistical methods, etc., or a combination thereof, [75]-[83]. Among methods that do not require segmentation, linear correlation filters have been both popular and successful, [84], [85]. These filters are inherently shift-invariant, and can be efficiently implemented either digitally or optically. On the other hand, multiple linear filters are required to account for wide variations of the target(s). In addition, each of the filters is usually synthesized separately leading to the computationally expensive and error prone task of searching multiple correlation planes independently. Recently,

ATR using Quadratic Correlation Filters (QCFs), received considerable attention, [86], [87]. These filters operate directly on image data without requiring segmentation or feature extraction, and retain the inherent shift-invariance of linear correlation filters. In addition, they considerably simplify the post-processing complexity required when using multiple linear correlation filters. The Rayleigh Quotient Quadratic Correlation Filter (RQQCF) technique was recently proposed that formulates the class separation metric as a Rayleigh quotient that is optimized by the QCF solution, [88], [89], [147]. As a result, the means of the two classes are well separated while simultaneously ensuring that the variance of each class is small. Chapter 2 of this dissertation gives a brief summary of the RQQCF method.

In the RQQCF method, the filter coefficients are obtained from the eigenvectors of a matrix computed from the autocorrelation matrices of targets and clutter. When new data is required to be added, these filter coefficients have to be updated which implies that the Eigenvalue Decomposition (EVD) has to be repeated. It is desirable to have methods that eliminate the need to perform an EVD every time the matrix changes but instead update the EVD adaptively, starting from the initial EVD. Although there is no paper in the literature that reports an adaptive algorithm specifically for the RQQCF method, there are many contributions that address the more general problem of adaptive eigendecomposition. This problem is known by many names, among which the most common and popular are “Adaptive Eigenvalue Decomposition”, “Subspace Tracking”, ‘Adaptive PCA” and “Adaptive Karhunen-Loeve Transform”. These methods find applications in many areas in signal processing, like Spectral Estimation, Source

Localization, Pattern Recognition, Wireless Communications, etc. All the algorithms reported, generally fall under one of the following four broad categories, [105], [108]:

- Classical batch EVD/SVD methods like QR algorithm, power iteration and Lanczos methods [77], [84], which are modified for adaptive processing, [99], [106], [107], [122].

- Rank-one updating algorithms like subspace averaging or reduced power iteration that find only some strong Eigenpairs, [109]-[113], [131], [145].

- Neural network based techniques that involve Hebbian or anti-hebbian learning, and lateral interaction, [96], [100], [101], [103], [117], [129], [130], [137], [138], [140]

- Methods that approach this as a problem of constrained or unconstrained optimization. [94], [95], [97], [98], [102], [104], [114], [115], [121], [123], [124], [133]-[136], [141]-[144],

Chapter 3 describes an approach that falls under the fourth category, i.e., constrained/unconstrained optimization. Among these, the most popular are the gradient-based algorithms like gradient descent, steepest descent, conjugate gradient, Newton-Raphson, and Recursive Least Squares (RLS). Each optimization method applied to a different objective function leads to a new algorithm.

The RQQCF technique operates on spatial domain data. Furthermore, each two-dimensional data chip in the spatial domain is converted into a one-dimensional vector by the lexicographical ordering of the columns of the chip. This leads to two interrelated issues. Firstly, the spatial structure in the two-dimensional chip is lost by converting it into a vector as described above.

Secondly, the dimensionality (size) of the system is increased considerably. One way to tackle both these issues simultaneously is to synthesize the RQQCF in the transform or frequency domain. Transforms capture the spatial correlation in images, and de-correlate the pixels. Consequently, if the transforms are appropriately selected, they compact the energy in the image in relatively few coefficients. Thus spatial domain data is transformed into an efficient and compact representation. Chapter 4 describes a transform domain formulation of the RQQCF and illustrates its advantages by sample simulation results on Infrared (IR) data for an ATR application.

Chapter 5 introduces the two dimensional (2D) RQQCF. In this approach, the aim is to reduce the computational complexity and storage requirements by keeping the dimensions of the target and clutter chips small. As opposed to the techniques described in Chapters 4 and 5, these target and clutter chips are not converted to vectors by lexicographical ordering of the columns at any stage. They are treated as 2D objects and the RQQCF formulation is appropriately changed. Sample results for a facial recognition/classification application illustrate the benefits and advantages of the proposed technique. Finally, Chapter 6 presents conclusions.

CHAPTER TWO: THE RAYLEIGH QUOTIENT QUADRATIC CORRELATION FILTER (RQQCF)

As mentioned in the previous chapter, the Rayleigh Quotient Quadratic Correlation Filter (RQQCF) technique improves discrimination between classes by explicitly optimizing a class separation metric. The metric is a ratio of two quantities – the numerator is the difference between the expected values of the filter outputs for the two classes and the denominator is an upper bound on the variance of the two classes. This chapter briefly reviews this technique and presents some results that can be used for comparison with techniques presented in later chapters.

The RQQCF Technique

In the RQQCF technique, the QCF coefficient matrix T is assumed to take the form,

$$T = \sum_{i=1}^{n} \underline{w_i}\,\underline{w_i^T} \tag{1}$$

where, $\underline{w_i}, 1 \le i \le n$, form an orthonormal basis set. The objective of the technique is to determine these basis functions such that the separation between the two classes, say X and Y, is maximized. The output of the QCF to an input vector u is given by

$$\varphi = \underline{u}^T T \underline{u} \tag{2}$$

The separation between the outputs when the inputs are from the target class, X and the outputs when the inputs are from the clutter class Y, is given by

$$E_1\{\varphi\} - E_2\{\varphi\} = E\{\underline{u_X}^T T \underline{u_X}\} - E\{\underline{u_Y}^T T \underline{u_Y}\} \tag{3}$$

where, $E_j\{.\}$ is the expectation operator over the jth class,

Substituting for T from Equation 1 into Equation 3,

$$E\ \{\varphi\} - E_2\{\varphi\} = \sum_{i=1}^{n} \underline{w_i}^T E\{\underline{u_X}\,\underline{u_X}^T\}\underline{w_i} - \sum_{i=1}^{n} \underline{w_i}^T E\{\underline{u_Y}\,\underline{u_Y}^T\}\underline{w_i}$$

$$= \sum_{i=1}^{n} \underline{w_i}(R_x - R_y)\underline{w_i^T} \tag{4}$$

where R_x and R_y are the correlation matrices for targets and clutter respectively.

An upper bound on the variance of the two classes is given by,

$$E\ \{\varphi\} + E_2\{\varphi\} = \sum_{k=1}^{n} u_X T u_X + \sum_{k=1}^{n} u_Y T u_Y$$

$$= \sum_{i=1}^{n} \underline{w_i}(R_x + R_y)\underline{w_i^T} \tag{5}$$

The objective is to maximize the ratio,

$$J(\underline{w}) = \frac{E_1\{\varphi\} - E_2\{\varphi\}}{E_1\{\varphi\} + E_2\{\varphi\}} = \frac{\sum_{i=1}^{n} \underline{w_i}(R_x - R_y)\underline{w_i^T}}{\sum_{i=1}^{n} \underline{w_i}(R_x + R_y)\underline{w_i^T}} \tag{6}$$

Taking the derivative of Equation 3 with respect to $\underline{w_i}$, and setting it to zero, we get

$$(R_x + R_y)^{-1}(R_x - R_y)\underline{w_i} = \lambda_i \underline{w_i} \tag{7}$$

Let,

$$A = (R_x + R_y)^{-1}(R_x - R_y) \tag{8}$$

Thus w_i is an eigenvector of A with eigenvalue λ_i . It should be noted that $J(\underline{w})$ is in the form of a Rayleigh Quotient which is maximized by the dominant eigenvector of A. The remaining *n-1* vectors are chosen from the *n-1* eigenvectors of A in order of decreasing eigenvalues. These *n* eigenvectors can now be used to construct the QCF using Equation 2.

In practice, *M* target and *M* clutter training sub-images, referred to as chips, are obtained from IR imagery. Each chip, having dimensions $\sqrt{n}$ x $\sqrt{n}$, is converted into a 1-D vector of dimensions *n x 1* by concatenating its columns. Target and clutter training sets of size *n x M* each, are obtained by placing the respective vectors in matrices. The *n x n* autocorrelation matrices of the target and clutter sets, R_x and R_y are computed, and used to obtain A according to Equation 5. As a result, the eigenvalues of A vary from –1 to +1. The dominant eigenvalues for clutter, λ_{ci} , are close to or equal to -1 and those for targets, λ_{ti} , are close to or equal to +1. The RQQCF coefficients, w_{ci} and w_{ti}, are mapped to the corresponding eigenvalues. In the original paper, the RQQCF is

correlated with an input scene to obtain a correlation surface from which the existence and location of the target is deduced. An efficient method to perform the correlation is discussed in the original paper, [88]. In this work though, to identify a data point as target or clutter, the sum of the absolute value of the k inner products of a data point with w_{ci} and w_{ti}, pt and pc, are calculated. If pt > pc, the data point is identified as a target. Otherwise, it is identified as clutter.

Simulation Results

Some sample simulation results are presented using Infrared (IR) data from Lockheed Martin MFC. The dataset consists of several video sequences of tanks and other vehicles on interest in various cluttered backgrounds. We have chosen four of these videos – VIDEO 1, VIDEO 2, VIDEO 3 and VIDEO 4 to demonstrate some of the results of the RQQCF. Figures 2.1 (a)-(d) show sample frames from these videos.

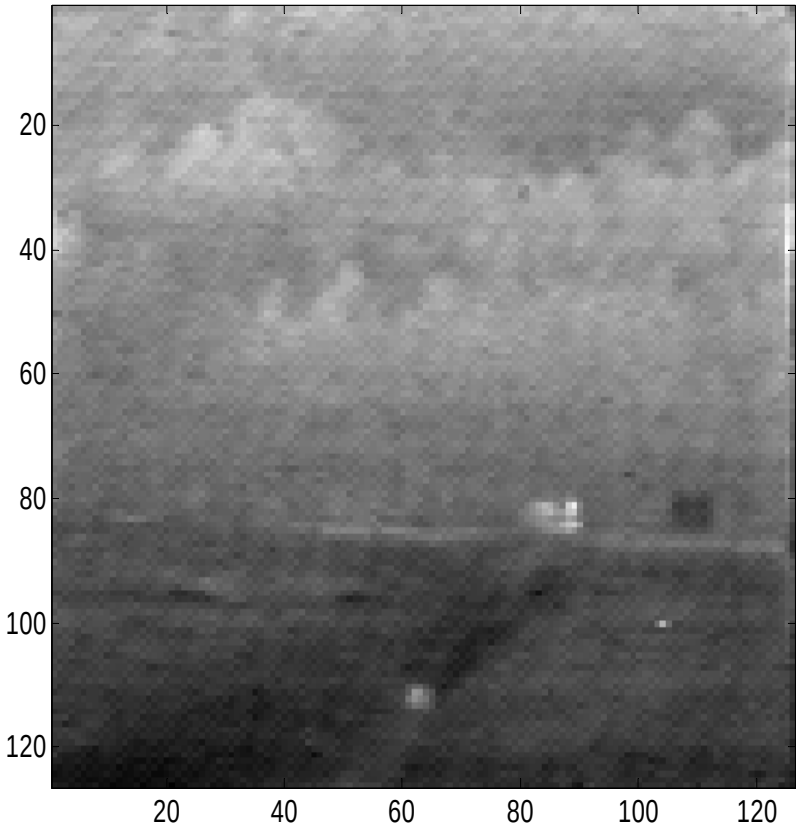

Figure 2.1(a) Sample frame from Video 1

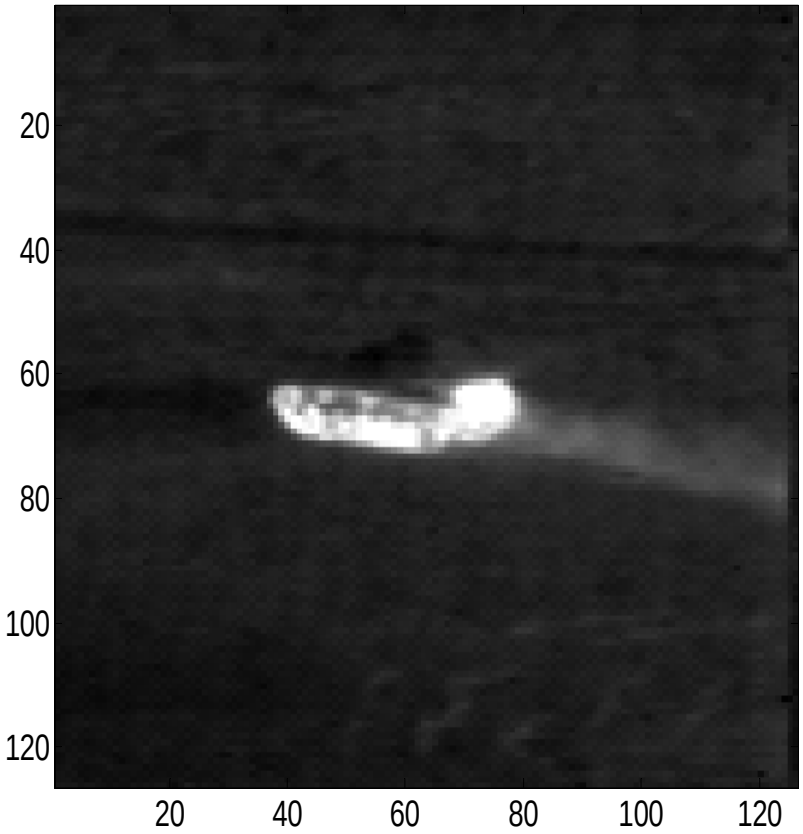

Figure 2.1(b) Sample frame from Video 2

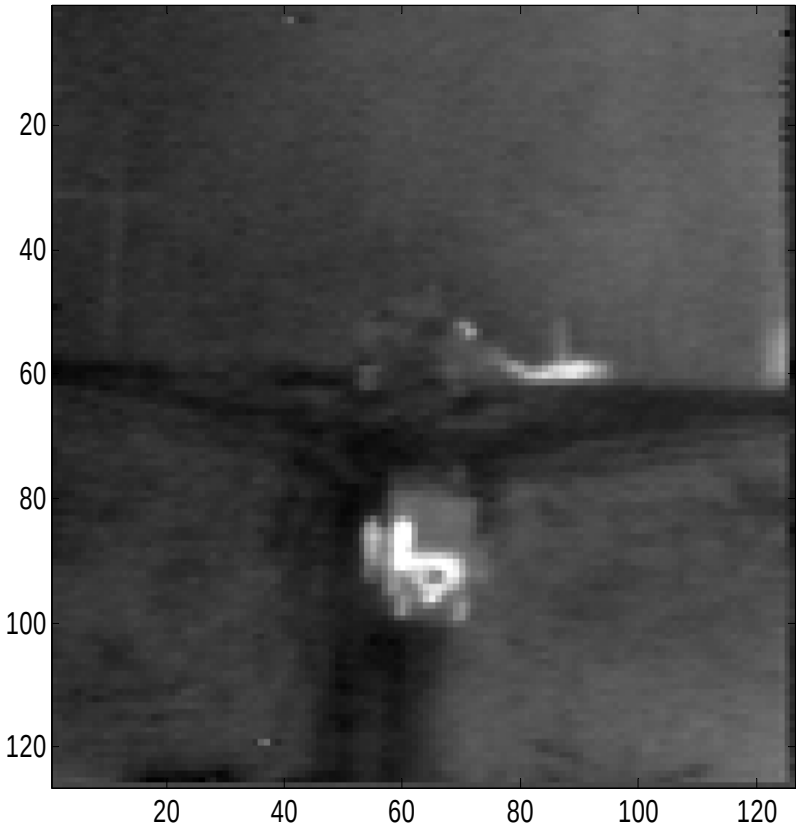

Figure 2.1(c) Sample frame from Video 3

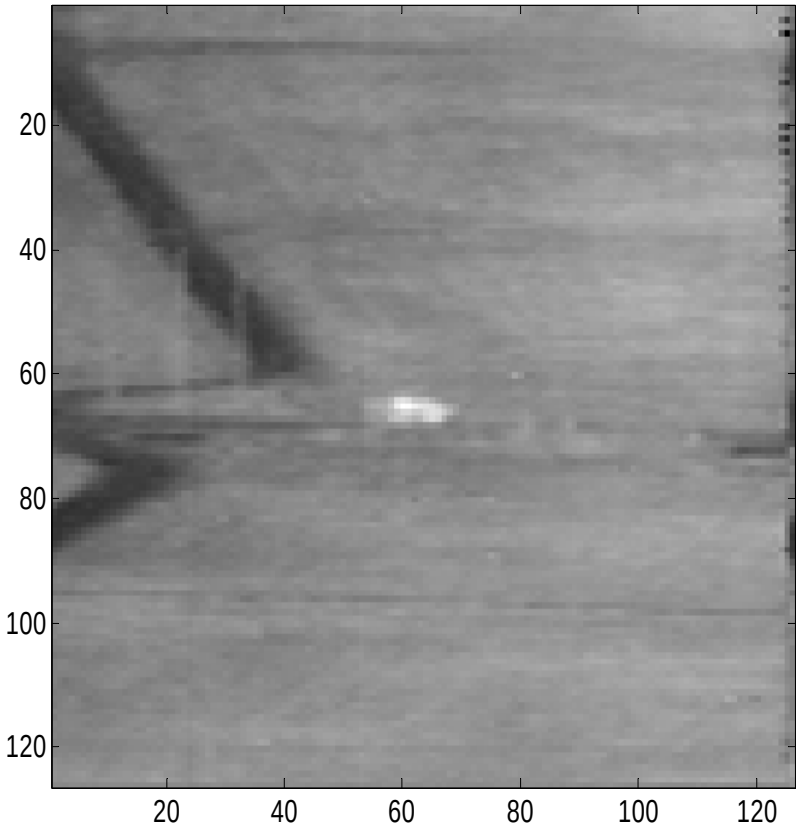

Figure 2.1(d) Sample frame from Video 4

Table 2.1 shows the number of frames in each video and the number of target and clutter chips, M, obtained for each video. Target chips are obtained from each frame of a video using ground truth data that is available. For clutter, chips are picked from all areas of each frame of the video except the area(s) where the target(s) is/are located. While this results in a larger number of clutter chips than target chips, for our simulations, the number of clutter chips is chosen to be equal to the number of target chips for convenience sake. Note that the size of the autocorrelation

matrices depends only on the dimension of the data points and not on the number of data points. The size of each chip is 16x16. Figure 2.2 shows the distribution of eigenvalues obtained for the RQQCF for VIDEO 1.

Table 2.1 Number of Frames and Number of Target and Clutter chips, M, for each video.

	VIDEO 1	VIDEO 2	VIDEO 3	VIDEO 4
Number of Frames	388	778	410	300
Number of Target and Clutter Chips, M	409	763	405	391

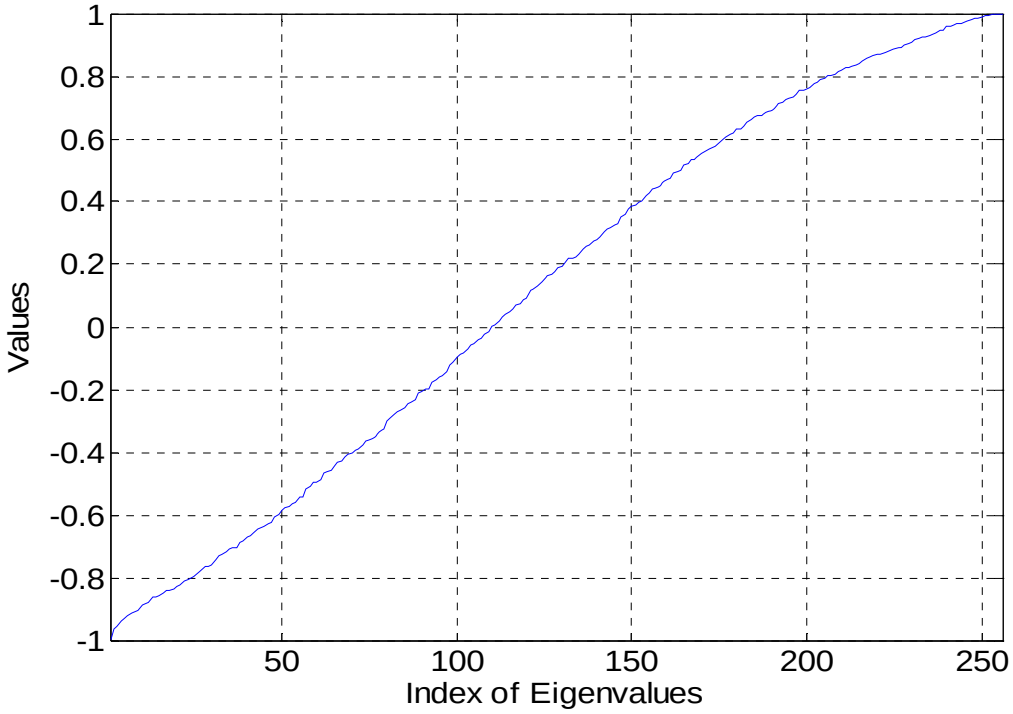

Figure 2.2 Distribution of eigenvalues corresponding to VIDEO 1

Figures 2.3 – 2.6 show the responses of representative target and clutter points from VIDEOS 1-4 respectively.

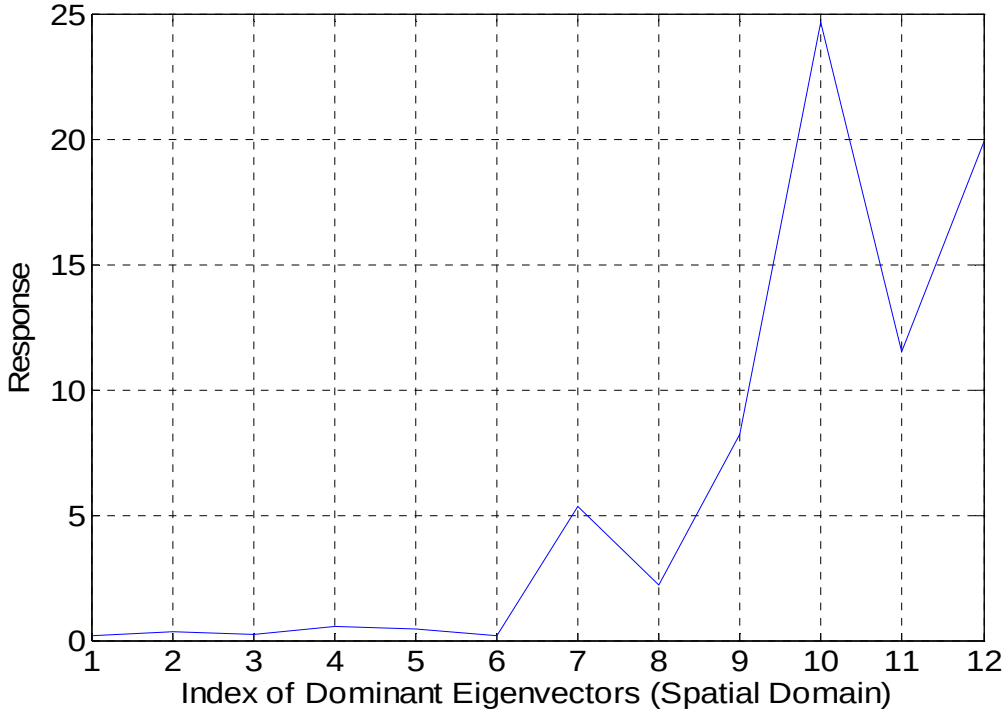

Figure 2.3(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors

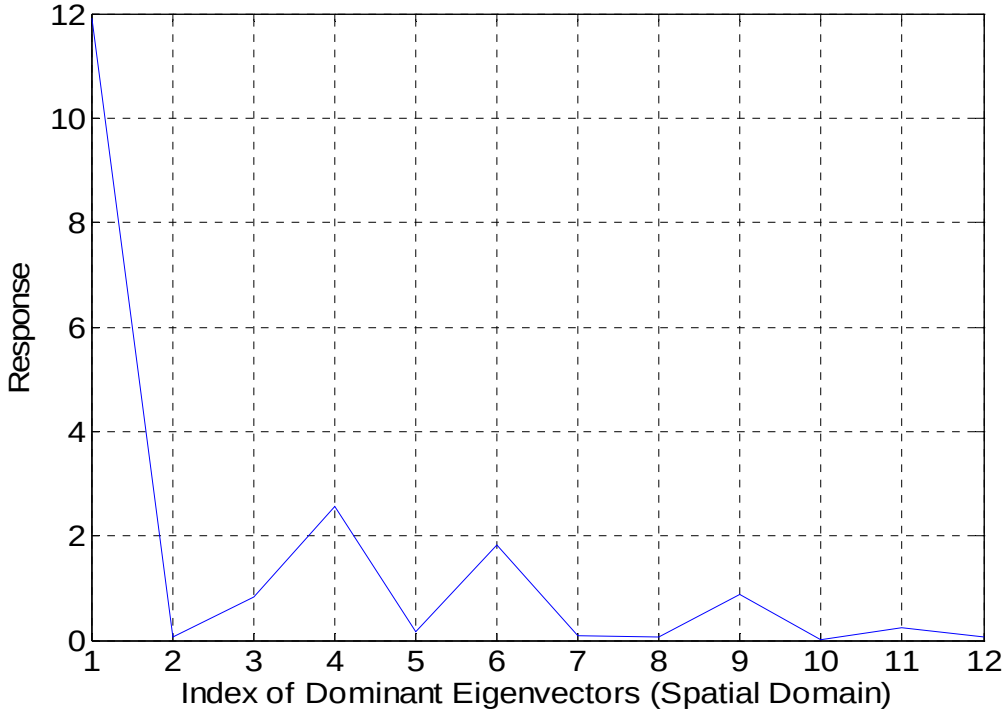

Figure 2.3(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors

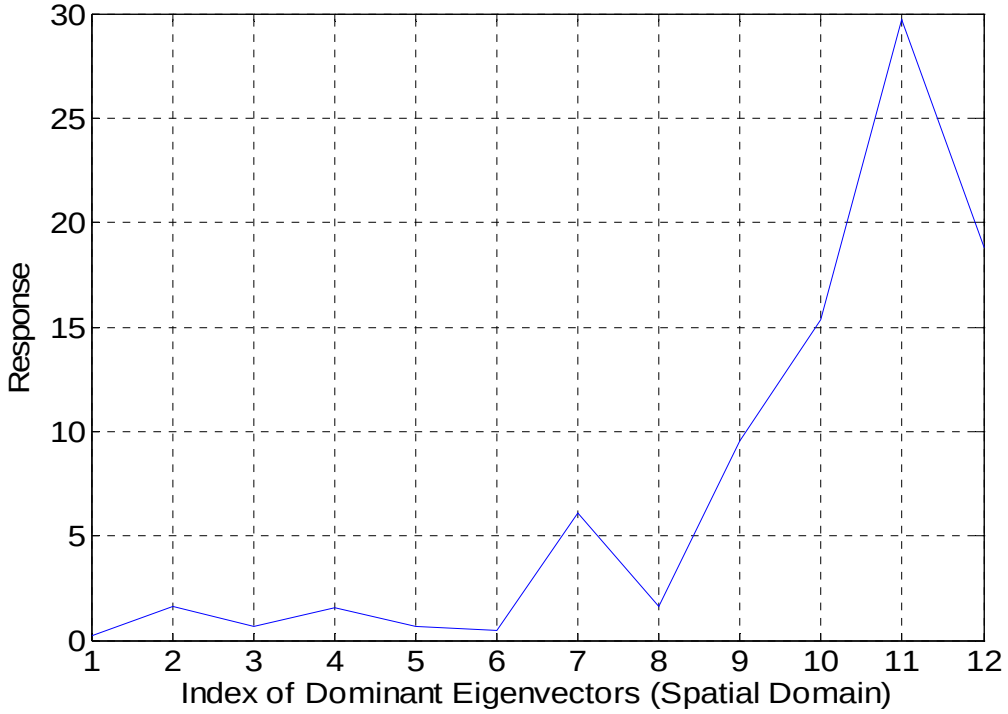

Figure 2.4(a) VIDEO 2: Response of a representative target vector versus the index of the dominant eigenvectors

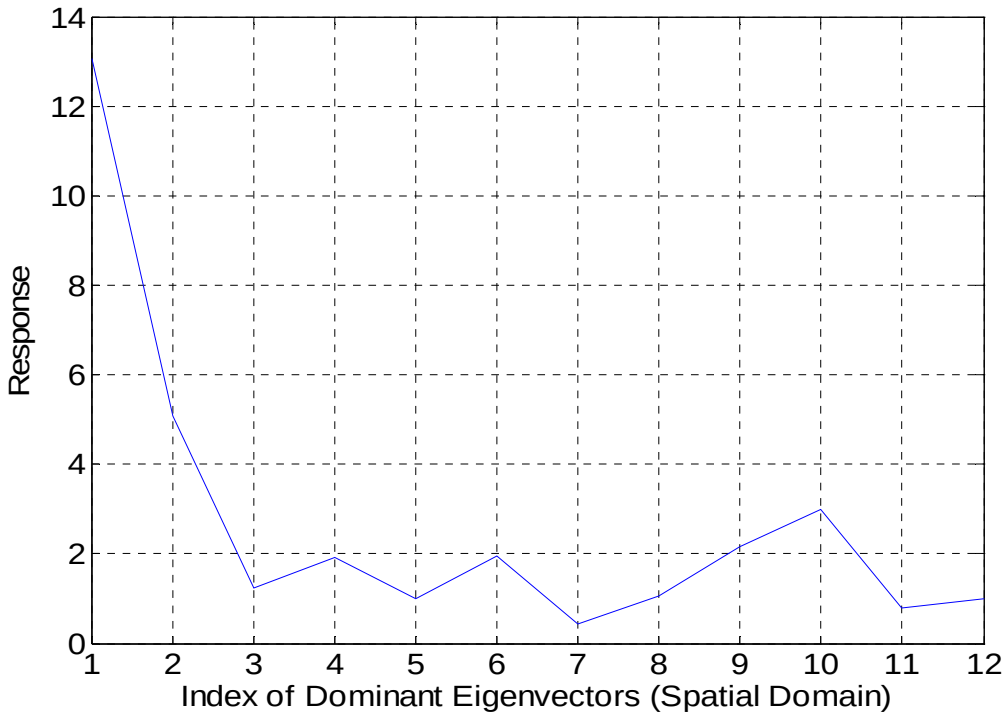

Figure 2.4(b) VIDEO 2: Response of a representative clutter vector versus the index of the dominant eigenvectors

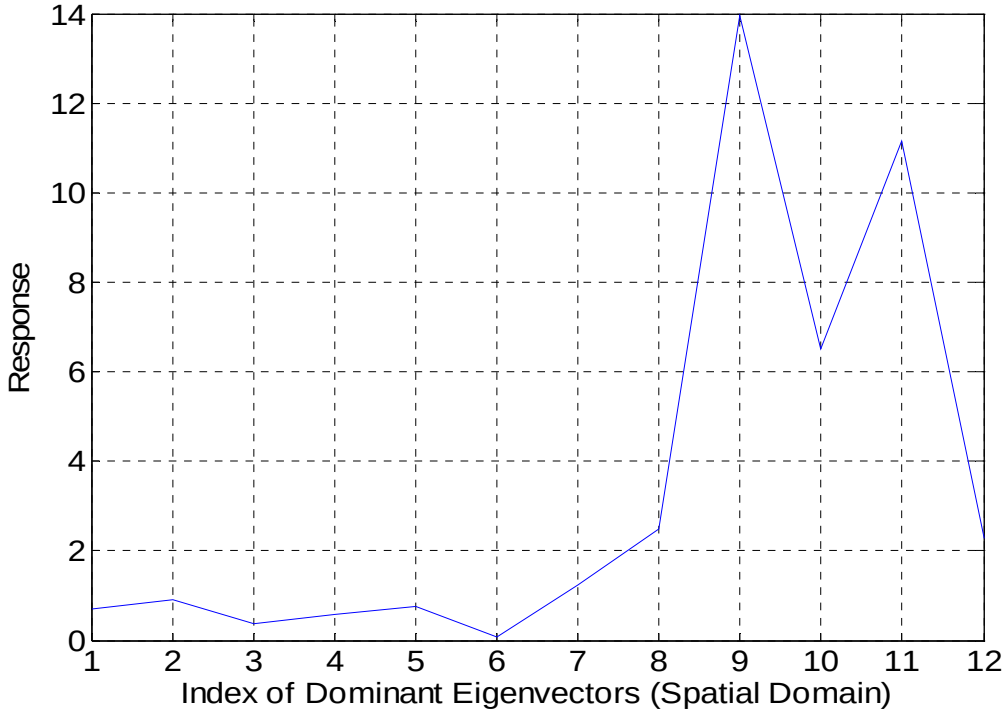

Figure 2.5(a) VIDEO 3: Response of a representative target vector versus the index of the dominant eigenvectors

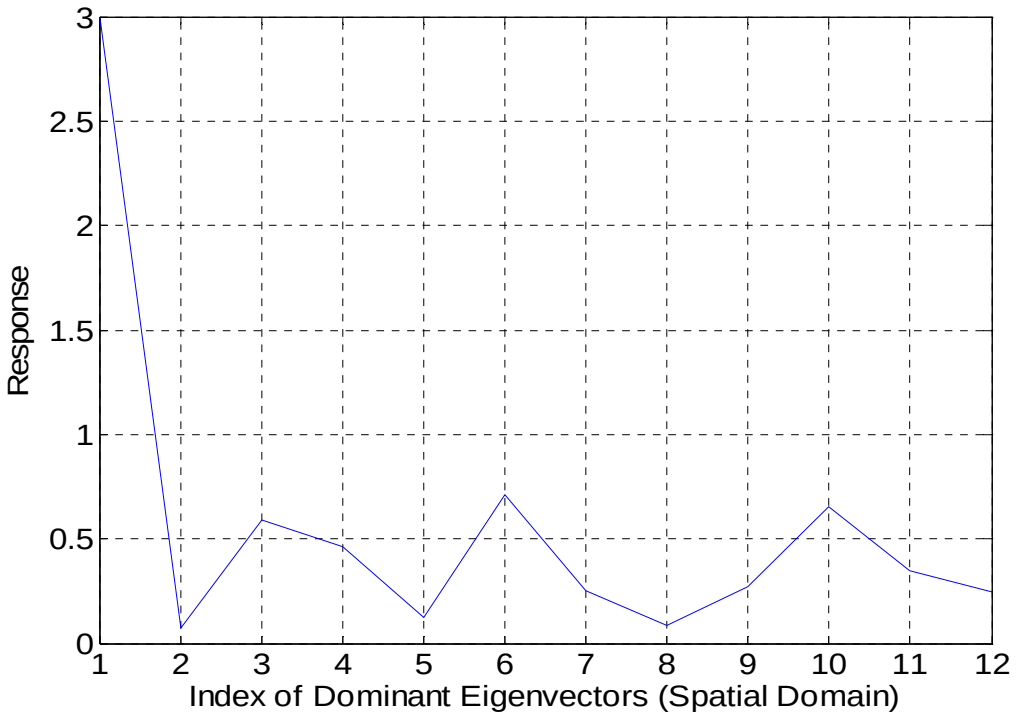

Figure 2.5(b) VIDEO 3: Response of a representative clutter vector versus the index of the dominant eigenvectors

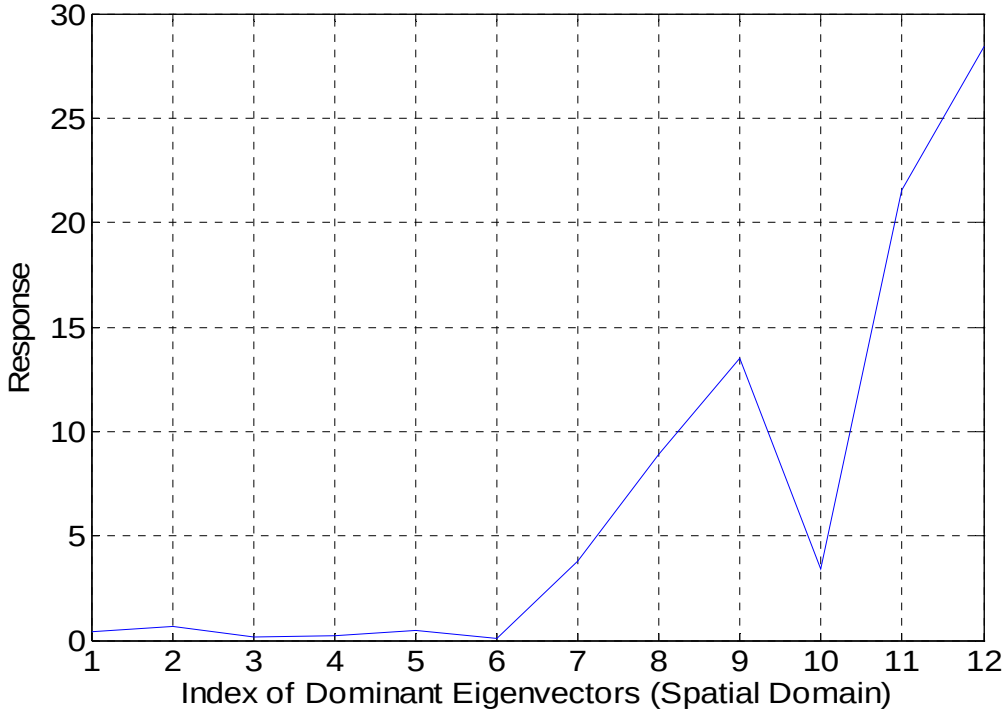

Figure 2.6(a) VIDEO 4: Response of a representative target vector versus the index of the dominant eigenvectors

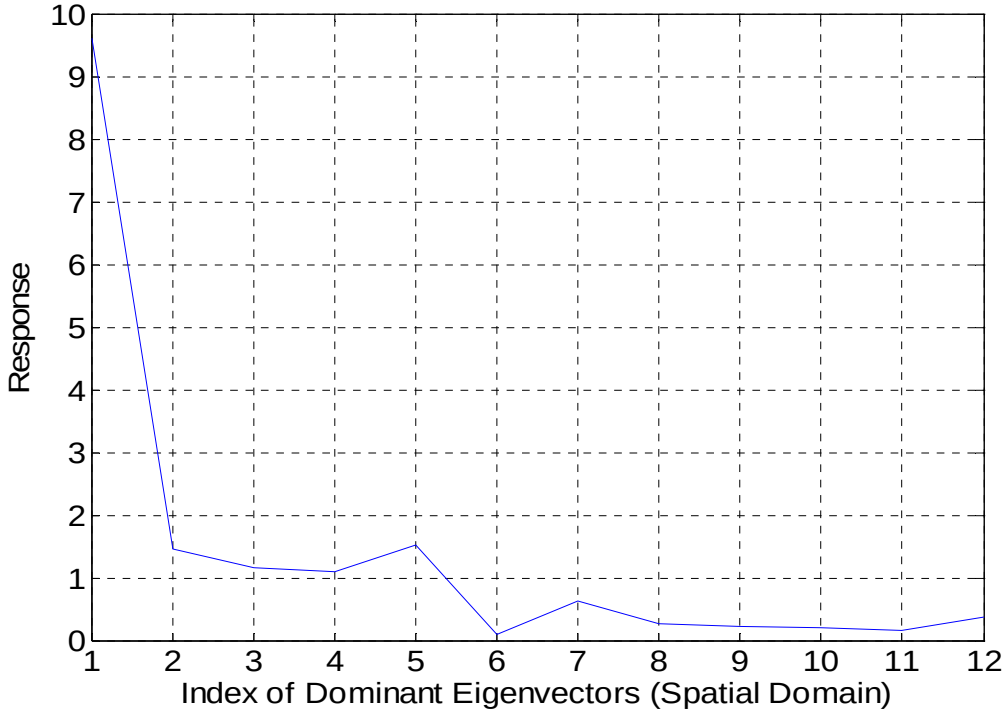

Figure 2.6(b) VIDEO 4: Response of a representative clutter vector versus the index of the dominant eigenvectors

Simulation results were performed for varying number of training and testing images. It was found that in general, the accuracy of the RQQCF was excellent for different scenarios where the training set size was varied.

Figures 2.7 – 2.10 show how the recognition rate, i.e. accuracy (in %) varies as a function of the number of training chips, i.e. the size of the training set, for the four videos, VIDEO 1 - 4. The

rates shown are averaged over several simulation runs, where the training set is picked from the available set randomly each time.

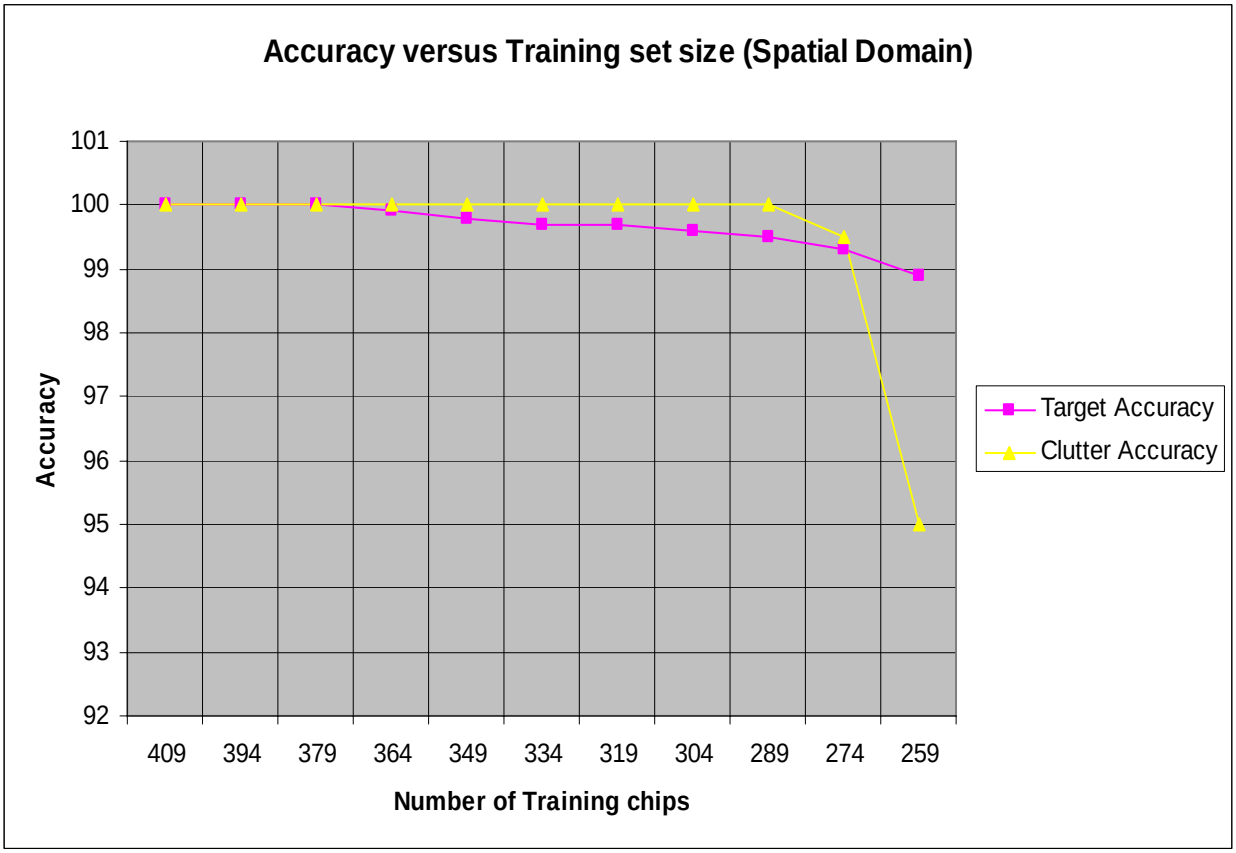

Figure 2.7 VIDEO 1: Accuracy (%) versus training set size

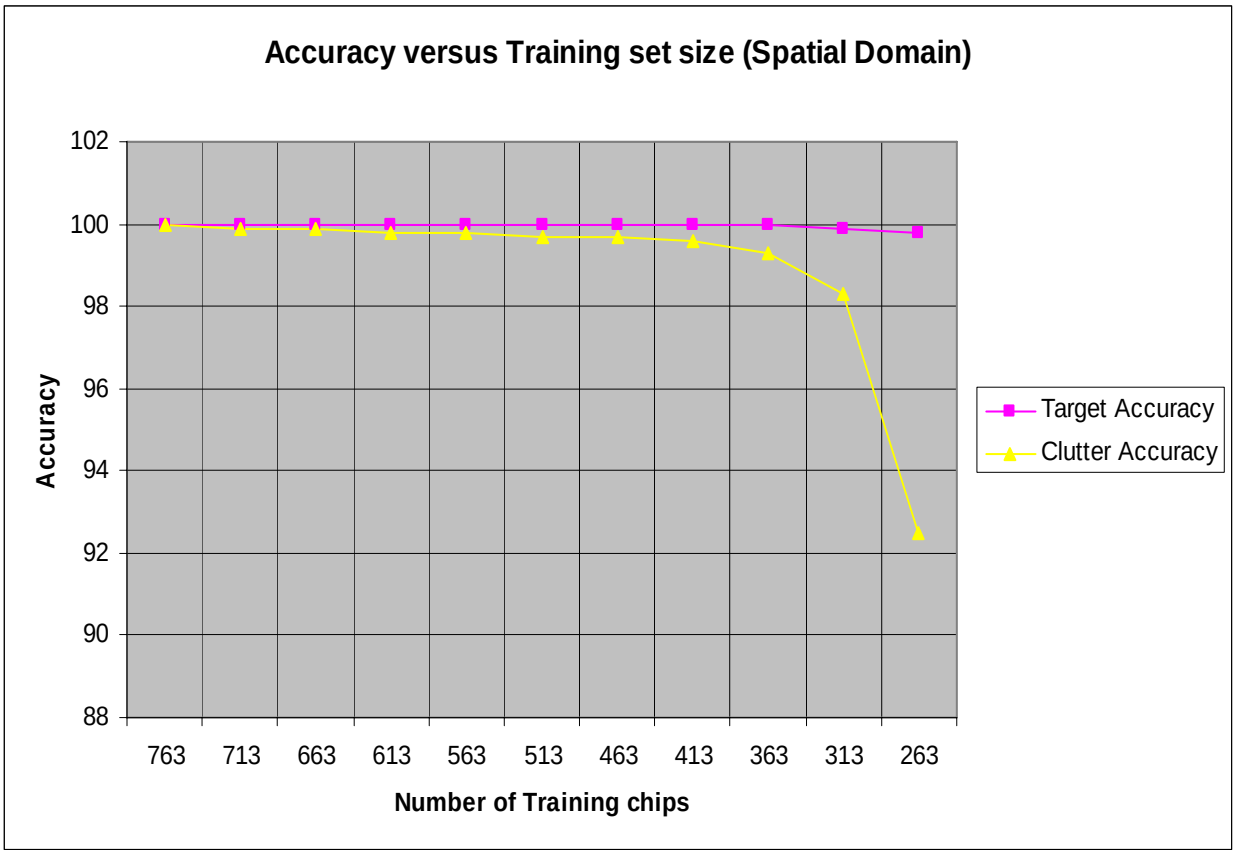

Figure 2.8 VIDEO 2: Accuracy (%) versus training set size

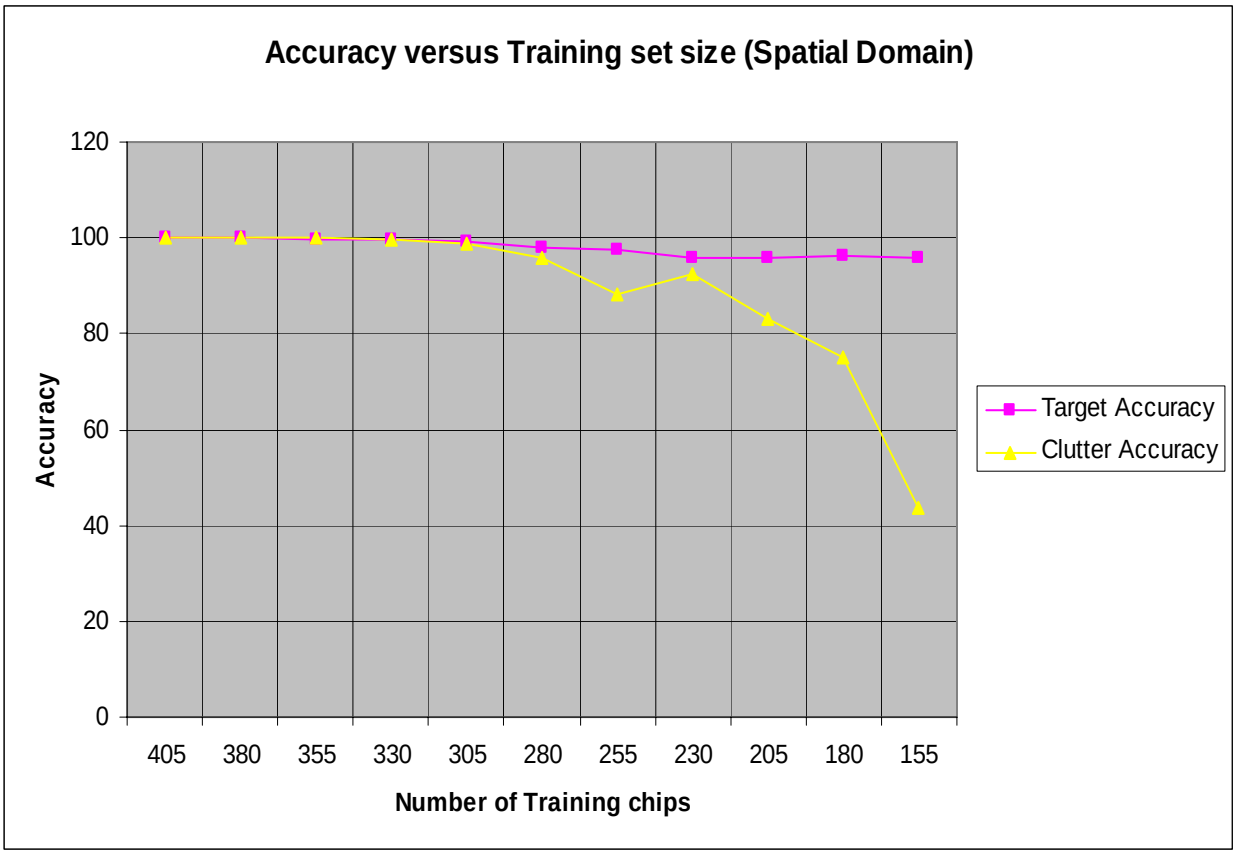

Figure 2.9 VIDEO 3: Accuracy (%) versus training set size

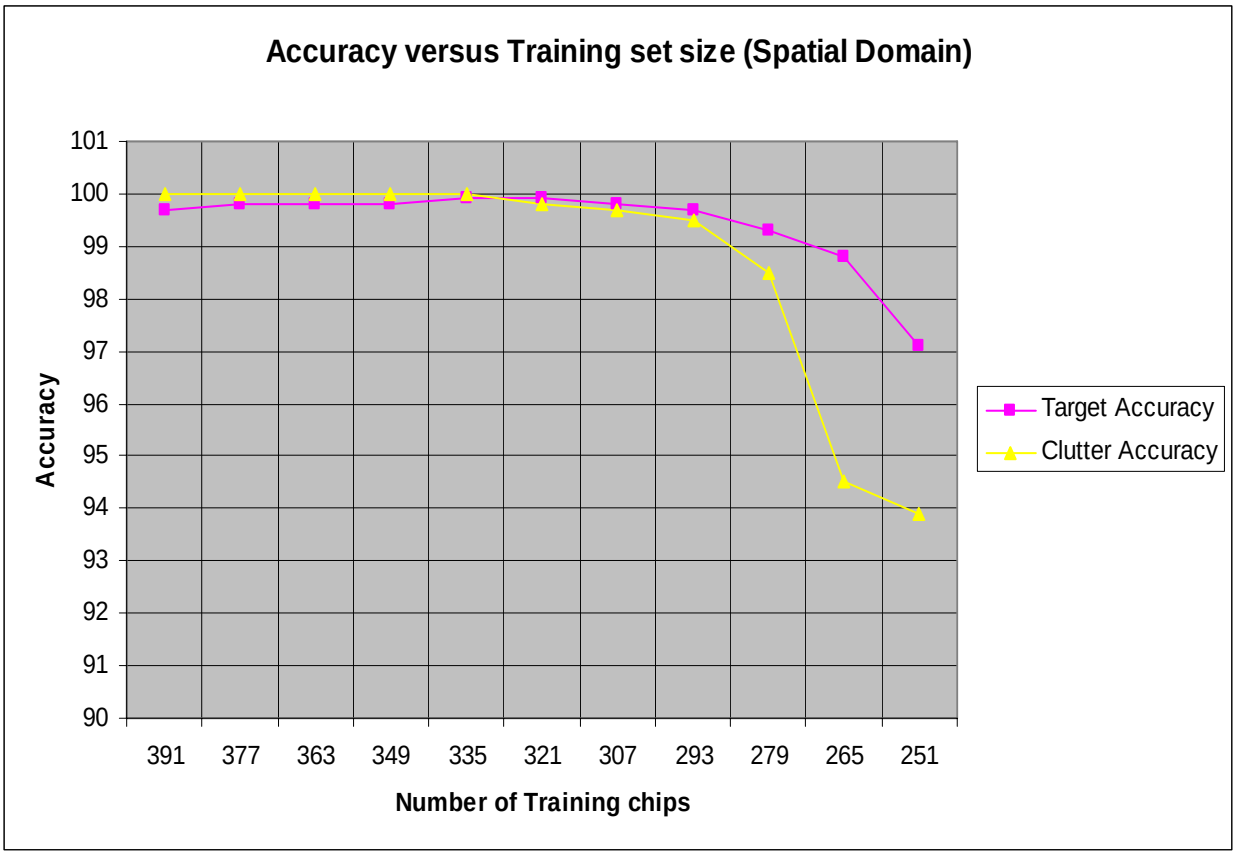

Figure 2.10 VIDEO 4: Accuracy (%) versus training set size

It is seen that the accuracy of the RQQCF is generally excellent, except when the size of the training set is close to or lesser than the size (dimension) of the data points themselves. This is to be expected because, as the size of the training set drops, the estimates of class statistics become poorer.

Summary

A brief summary of the Rayleigh Quotient Quadratic Correlation Filter (RQQCF) technique was presented. The technique is an elegant way of optimizing a bank of linear filters simultaneously to give a single response (correlation) surface from which decisions can be made. It requires no explicit segmentation and is very robust in even low contrast images. It has excellent recognition performance overall when the number of training and testing images are varied.

CHAPTER THREE: AN ADAPTIVE ALGORITHM FOR THE RQQCF

The QCF filter coefficients obtained using the RQQCF technique described in the previous chapter, are based on a training set of targets and clutter. The problem arises when the system is required to recognize new targets or operate in a new environment. In such cases, the target or clutter set has to be updated and in turn, the EVD. Undoubtedly, it is highly desirable to avoid solving the EVD problem from scratch again, since it may incur a large amount of computation. In this chapter, an adaptive technique called the Optimal Adaptive Eigenvalue Decomposition (OAEVD) is proposed that utilizes the old EVD to search for the new EVD. The technique avoids matrix inversion and direct EVD, thus providing substantial computational savings. In addition, it eliminates the need for storing old target and clutters sets, and allows us to solve only for as many eigenvalues and corresponding eigenvectors as desired, for e.g., the most significant eigenvalues and corresponding eigenvectors. Computer simulations confirm the effectiveness of the proposed technique.

OAEVD - Formulation

The perturbed EVD problem in the RQQCF involves decomposing the time-varying matrix,

$$A = (R_x + R_y)^{-1}(R_x - R_y) \tag{9}$$

where, R_x and R_y are the autocorrelation matrices of the target and the clutter sets respectively, which change with the addition of new targets or new clutter or both.

The first step in our adaptive formulation is to identify a "Cost Function" to be minimized when new data is incorporated. According to Equation 9, we choose the following as our "Error Signal":

$$\underline{e(j)} = [R_x - R_y - \lambda_i(R_x + R_y)]\underline{w_i(j)} \tag{10}$$

where, $\underline{e(j)}$ is an M by 1 vector, λ_i and $\underline{w_i(j)}$ are the ith Eigenvalue and the corresponding Eigenvector respectively. So the cost function is $\underline{e^T(j)}\underline{e(j)}$, which is the energy in our error signal.

Since there are two variables in the error signal, namely, λ_i and $\underline{w_i(j)}$, a simple way to tackle this adaptive problem is to keep one variable constant in one iteration while updating the other, and then vice versa, i.e., we use the updated $\underline{w_i(j)}$, to update λ_i during the same iteration.

Assuming that the changes in λ_i and $\underline{w_i(j)}$ are small, we derive the Taylor series expansion for $\underline{e(j+1)}$ in terms of $\underline{e(j)}$ and its partial derivatives with respect to $\underline{w_i(j)}$ and λ_i:

$$e_l(j+1) = e_l(j) + \sum_{k=1}^{M} \frac{\partial e_l(j)}{\partial w_{i,k}(j)} \Delta w_{i,k}(j) + \frac{\partial e_l(j)}{\partial \lambda_i(j)} \Delta \lambda_i(j) \tag{11}$$

where, l = 1, …, M. Writing (11) for l = 1, …, M:

$$\underline{e(j+1)} = \underline{e(j)} + [S_1 - \lambda_i S_2]\underline{\Delta w_i(j)} - S_2 \underline{w_i(j)}\Delta\lambda_i(j) \tag{12}$$

where, $S_1 = R_x - R_y; S_2 = R_x + R_y$.

Now, to update λ_i, $\underline{w_i(j)}$ is kept constant. Therefore, Equation 12 becomes

$$\underline{e(j+1)} = \underline{e(j)} - S_2 \underline{w_i(j)}\Delta\lambda_i(j) \tag{13}$$

To update $\underline{w_i(j)}$, λ_i is kept constant. In this case, Equation 12 becomes

$$\underline{e(j+1)} = \underline{e(j)} + [S_1 - \lambda_i S_2]\underline{\Delta w_i(j)} \tag{14}$$

In addition, we incorporate another constraint on $\Delta\underline{w_i(j)}$ and $\Delta\lambda_i$ - they should be proportional to the negative gradient of the cost function in the j[th] iteration, $\underline{e^T(j)}\underline{e(j)}$, with respect to $\underline{w_i(j)}$ and λ_i, respectively. Therefore,

$$\Delta\lambda_i(j) = -k_{\lambda_i} \frac{\partial[\underline{e^T(j)}\underline{e(j)}]}{\partial\lambda_i(j)} = -2k_{\lambda_i} \underline{w_i^T(j)} S_2[\lambda_i S_2 - S_1]\underline{w_i(j)} \tag{15}$$

$$\Delta w_i(j) = -[MU]_j \frac{\partial[\underline{e^T(j)}\underline{e(j)}]}{\partial\underline{w_i(j)}} = -\frac{2}{M}[MU]_j[S]\underline{e(j)} \tag{16}$$

In Equation 16, $S = S_1 - \lambda_i S_2$, and

$$[MU]_j = \begin{bmatrix} \mu_{B1}(j) & ..... & 0 \\ ..... & ..... & ..... \\ 0 & ..... & \mu_{BM}(j) \end{bmatrix} \tag{17}$$

Substituting Equation 15 into Equation 13, we can write

$$\underline{e^T(j+1)}\,\underline{e(j+1)} = [\underline{e^T(j)} + 2k_{\lambda_i}\underline{V^T(j)}][\underline{e(j)} + 2k_{\lambda_i}\underline{V(j)}] \tag{18}$$

This is a quadratic function of k_{λ_i}, so the optimum k_{λ_i} is given by:

$$k_{\lambda_i} = -0.5\frac{\underline{V^T(j)}\,\underline{e(j)}}{\underline{V^T(j)}\,\underline{V(j)}} \tag{19}$$

where,

$$\underline{V^T(j)} = S_2\,\underline{w_i(j)}[\lambda_i\,\underline{w_i^T(j)}S_2 - \underline{w_i^T(j)}S_1]S_2\,\underline{w_i(j)} \tag{20}$$

Substituting Equation 19 into Equation 15, we obtain the update equation for $\Delta\lambda_i$. To update $\underline{w_i(j)}$, substitute Equation 16 into Equation 14, we can write,

$$\underline{e^T(j+1)}\,\underline{e(j+1)} = A_1 + A_2 + A_3 \tag{21}$$

where,

$$A_1 = \underline{e^T(j)}\,\underline{e(j)}, \tag{22}$$

$$A_2 = -\frac{4}{M}\underline{q^T(j)}[MU]_j\,\underline{q(j)} \tag{23}$$

$$A_3 = \frac{4}{M^2}\underline{q^T(j)}[MU]_j[R]_j[MU]_j\,\underline{q(j)} \tag{24}$$

where, $\underline{q(j)} = [S]\underline{e(j)}$ and $[R]_j = [S]^2$.

Taking the derivative of Equation 21 with respect to each $\mu_{Bl}(j)$, and setting it to zero:

$$[MU]_j^*\,\underline{q(j)} = \frac{M}{2}[R]_j^{-1}\,\underline{q(j)} \tag{25}$$

Substituting Equation 25 into Equation 16,

$$\underline{\Delta w_i(j)} = -[R]_j^{-1}\,\underline{q(j)} \tag{26}$$

Therefore, the final update equations are:

$$\Delta\lambda_i(j) = \frac{\underline{V^T(j)}\,\underline{e(j)}}{\underline{V^T(j)}\,\underline{V(j)}}\,\underline{w_i^T(j)}S_2[\lambda_i S_2 - S_1]\underline{w_i(j)} \tag{27}$$

$$\underline{\Delta w_i(j)} = -[R]_j^{-1}\,\underline{q(j)} \tag{28}$$

The above algorithm avoids the EVD required for the new A matrix, but it still needs the inverse of [R], which is of the same dimensionality as the data points. In fact, we notice that the above

solution can be obtained equivalently by setting $\underline{e(j+1)} = 0$ in Equation 13 and Equation 14. Thus, this is not a truly iterative solution.

To obtain a truly adaptive solution, direct matrix inversion has to be avoided. To this end, we approximate the inverse of matrix [R] by a diagonal matrix containing the reciprocals of [R]'s diagonal elements. Additionally, a convergence factor μ is introduced in the update equations to ensure reliable convergence.

Thus the update equations become:

$$\Delta\lambda_i(j) = \mu \frac{V^T(j)e(j)}{V^T(j)V(j)} \underline{w_i^T(j)} S_2[\lambda_i S_2 - S_1]\underline{w_i(j)} \tag{29}$$

$$\underline{\Delta w_i(j)} = -\mu[B]\underline{q(j)} \tag{30}$$

where, $[B] = diag[1/diag[R]]$.

OAEVD - Simulation Results for Synthetic Data

The adaptive algorithm was tested using various 1D and 2D synthetic datasets. Sample results each for three of the datasets is presented. The algorithm was found to be successful in extensive simulations.

Dataset 1: Dataset 1 consists of strings of binary digits, 15 bits long. A pattern of 3 bits of the form "1 1 1" is considered a target. The target set consists of all 15 bit strings containing a "1 1 1" with the pattern placed in a different position in each string. All other bits are set to zero. For example, "1 1 1 0 0 0 0 0 0 0 0 0 0 0 0", is a string in the target set. A pattern of 3 bits of the form "0 1 0" is considered clutter. The clutter set consists of all 15 bit strings containing a "0 1 0" with the pattern placed in a different position in each string. All other bits set to zero. For example, "0 1 0 0 0 0 0 0 0 0 0 0 0 0 0", is a string in the clutter set. Thus, we obtain 13 combinations each for the target and clutter sets. To solve potential rank deficiency problems with the dataset, data points corrupted with Gaussian random noise of small magnitude are also incorporated into both target and clutter sets. As a result, the target and clutter data sets now have dimensionality of 26 x 15. Thus, M, n and k are 26, 15, and 15 respectively The new data point to be incorporated is chosen to be a reasonable variation of the strings used in the training set, for example, "1 0 0 0 0 0 0 0 0 0 0 0 0 1 1".

Dataset 2: In Dataset 2, the target set consists of 512 discrete sinusoids each of length 64. Each data point is a sinusoid of a different frequency. The frequency of the sinusoids increases from

the first data point to the last. The clutter set consists of 512 data points, each of which is a vector of random noise having a Gaussian distribution. The first 128 points of the target and clutter set are used to form the initial autocorrelation matrices. The remaining points are used as new data points. . Thus, M, n and k are 128, 64, and 10 respectively

Dataset 3: For the target set, chips of size 5x5 each containing random noise with a Gaussian distribution are generated. A string of the form "1 1 1", considered a target, is embedded in a different position in each chip to obtain different target chips. For the clutter set, another set of chips containing random noise with a Gaussian distribution are generated. Thus, we obtain thirty target and thirty clutter data points. The first twenty-five data points in each set are used to form the initial autocorrelation matrices. The remaining points are used as new data. Figure 3.1 and Figure 3.2 show sample target and clutter chips, respectively. As explained in Section 2, these two-dimensional chips are converted into one-dimensional vectors, each of dimensions 25x1. Thus, M, n and k are 25, 25, and 10 respectively

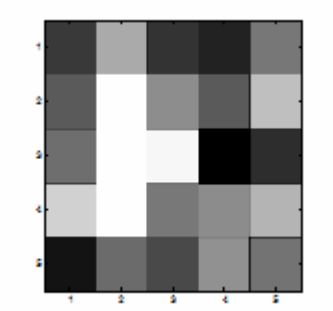

Figure 3.1 Sample target chip from Dataset3

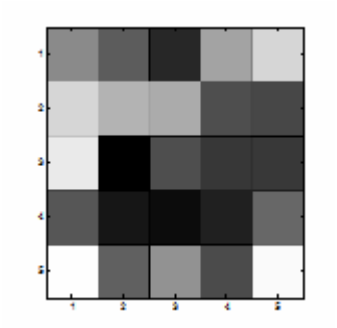

Figure 3.2 Sample clutter chip from Dataset3

For each of the datasets, the matrix A is computed according to Equation 1. The EVD of A is performed to obtain the initial λ_i's and $\underline{w_i}$'s. Then, the new data points are added and the adaptive algorithm is used to track the changes in the λ_i's and $\underline{w_i}$'s. For the sake of evaluating the new algorithm, the $\hat{\lambda}_i$'s and $\underline{\hat{w}_i}$'s obtained from the OAEVD algorithm are compared with the exact values λ_{ie}'s and w_{ie}'s obtained by recalculating A, and performing the actual EVD. Adaptation is terminated when the change in the eigenvectors from one iteration to the next falls below a certain threshold ε. All simulations are performed using MATLAB 7.

Dataset 1: A new data point 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1" is added to the initial set of data points. To make $\Delta\lambda_i$ reasonably small, the new data point is de-emphasized by introducing a scaling factor. In the experiment, it is found that 10^{-4} is a good scaling factor. The new perturbed target autocorrelation matrix is obtained. The OAEVD is used to calculate the $\hat{\lambda}_i$'s and $\underline{\hat{w}_i}$'s. All the 15

target and clutter eigenpairs are computed. μ and ε used are 0.1 and 1e-15, respectively. The result is summarized below.

λ_i = -1.0000, -0.9999, -0.7446, -0.6774, -0.2619, -0.2580, -0.0412, 0.1637, 0.4554, 0.6226, 0.7150, 0.7668, 0.7918, 1.0000, 1.0000

λ_{ie} = -1.0000, -0.9999, -0.7529, -0.6874, -0.2794, -0.2755, -0.0601, 0.1452, 0.4403, 0.6109, 0.7056, 0.7589, 0.7847, 1.0000, 1.0000

$\hat{\lambda}_i$ = -1.0000, -0.9999, -0.7529, -0.6874, -0.2794, -0.2755, -0.0601, 0.1452, 0.4403, 0.6109, 0.7056, 0.7589, 0.7847, 1.0000, 1.0000

iter = 2, 1, 473, 331, 879, 685, 316, 891, 381, 353, 394, 452, 586, 1, 6

In the sample result, it was found that the eigenvectors, obtained using the OAEVD algorithm, $\underline{\hat{w}_i}$, were identical to w_{ie}.

Dataset 2: One new data point is added to the initial target set of 128 points. The new perturbed target autocorrelation matrix is obtained. The OAEVD is used to calculate the $\hat{\lambda}_i$'s and $\underline{\hat{w}_i}$'s. Five dominant target and clutter eigenpairs are computed. μ and ε used are 0.1 and 1e-25, respectively. The result is summarized below.

λ_i = -1.0000, -0.9999, -0.9986, -0.9892, -0.8723, 0.5821, 0.6518, 0.6685, 0.7140, 0.7468

λ_{ie} = -1.0000, -0.9999, -0. 9983, -0.9892, -0.8711, 0.5806, 0.6510, 0.6663, 0.7123, 0.7453

$\hat{\lambda}_i$ = -1.0000, -0.9999, -0.9983, -0.9892, -0.8711, 0.5806, 0.6510, 0.6663, 0.7123, 0.7453

iter = 1, 5, 2, 51, 29, 562, 426, 459, 730, 579

In the sample result, it was found that the eigenvectors, obtained using the OAEVD algorithm, were successfully able to distinguish between target and clutter. Please note that the eigenvectors, $\underline{w_i}$, $\underline{\hat{w}_i}$ & w_{ie} are not given because of space constraints. Figure 3.3 and Figure 3.4 show plots of the absolute value of the inner product of the new data point that was added, with $\underline{\hat{w}_i}$ and $\underline{w_{ie}}$, respectively. There is a good match between the two plots.

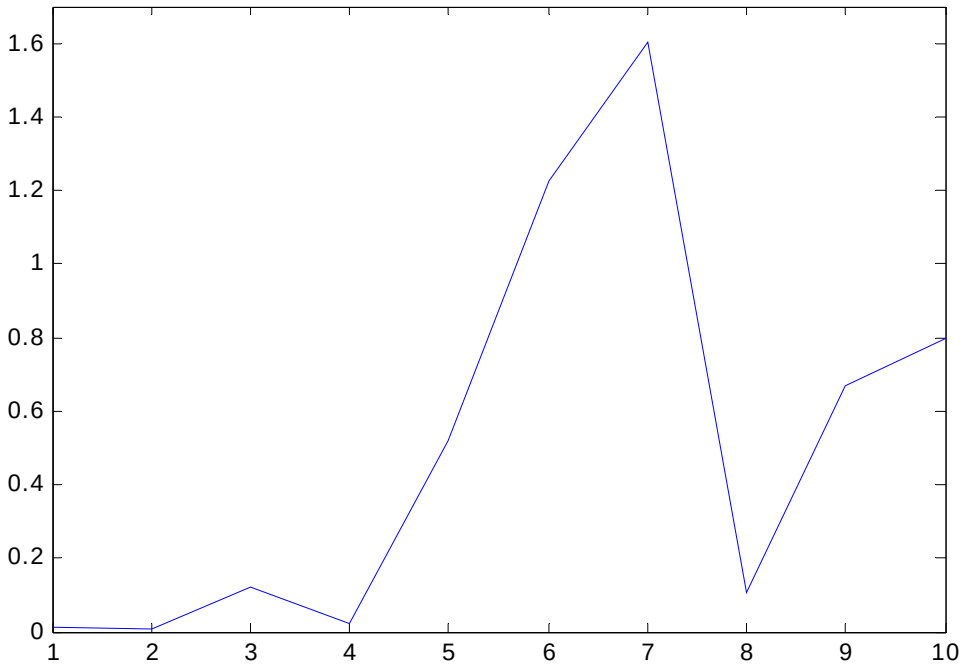

Fig 3.3 (Dataset 2) Absolute value of inner product of the new target vector with $\underline{\hat{w}_i}$ versus i, the index of the Eigenvectors. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

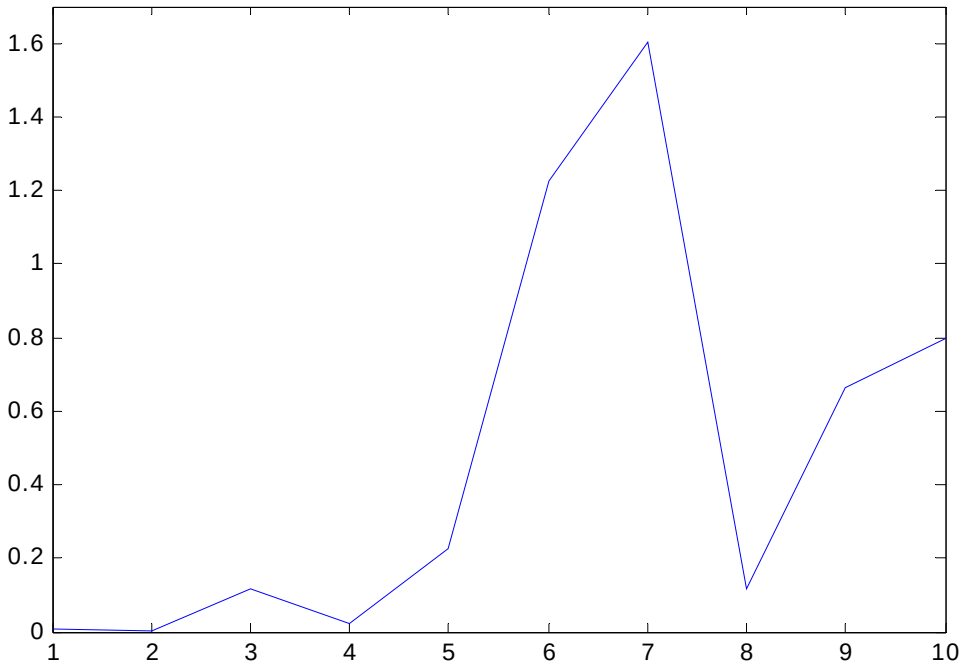

Figure 3.4 (Dataset 2) Absolute value of inner product of the new target vector with $\underline{w_{ie}}$ versus i, the index of the Eigenvectors. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

Dataset 3: One new data point is added to the initial target set of 25 points. The new perturbed target autocorrelation matrix is obtained. The OAEVD is used to calculate the $\hat{\lambda}_i$'s and $\underline{\hat{w}_i}$'s. Five dominant target and Five dominant clutter eigenpairs are computed. μ and ε used are 0.1 and 1e-25, respectively. The result is summarized below.

λ_i = -0.9772, -0.9651, -0.8899, -0.8774, -0.8317, 0.8555, 0.9459, 0.9482, 0.9710, 0.9997

λ_{ie} = -0.9768, -0.9340, -0.8932, -0.8617, -0.6847, 0.8531, 0.9444, 0.9463, 0.9719, 0.9997

$\hat{\lambda}_i$ = -0.9768, -0.9340, -0.8932, -0.8619, -0.6745, 0.8530, 0.9444, 0.9463, 0.9719, 0.9997

iter = 132, 237, 107, 156, 261, 65, 123, 172, 12

In the sample result, it was found that the eigenvectors, obtained using the OAEVD algorithm, were successfully able to distinguish between target and clutter. Please note that the eigenvectors, $\underline{w_i}$, $\underline{\hat{w}_i}$ & w_{ie} are not given because of space constraints. Figure 3.5 and Figure 3.6 show plots of the absolute value of the inner product of the new data point that was added, with $\underline{\hat{w}_i}$ and $\underline{w_{ie}}$, respectively. There is a good match between the two plots. Figure 3.7 and Figure 3.8 show the corresponding plots for a clutter point randomly selected from the training set.

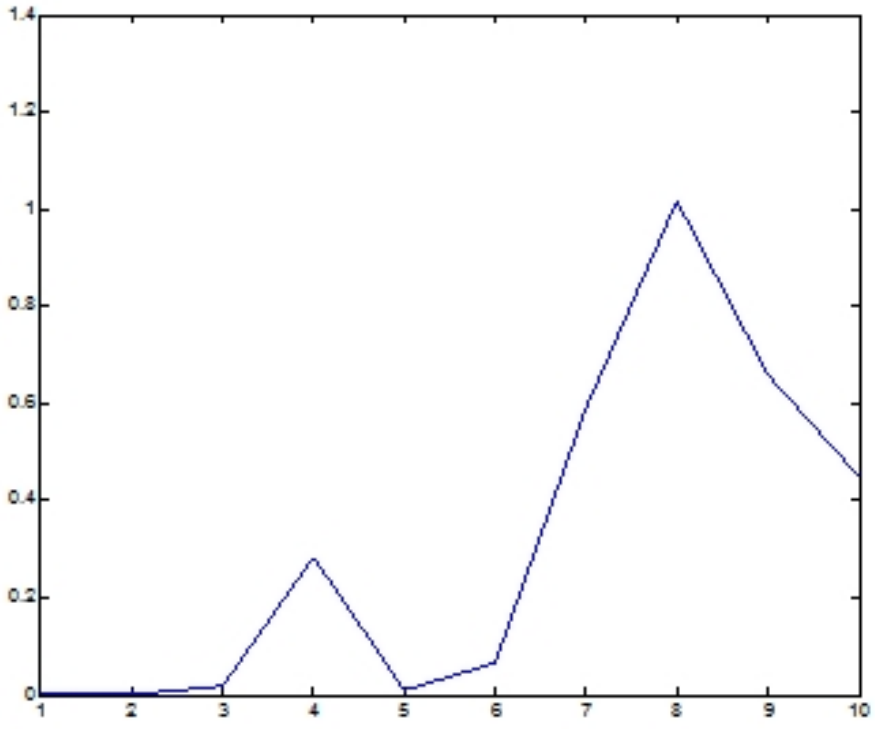

Figure 3.5 (Dataset 3) Absolute value of inner product of the new target vector with $\underline{\hat{w}_i}$ versus i, the index of the Eigenvectors. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

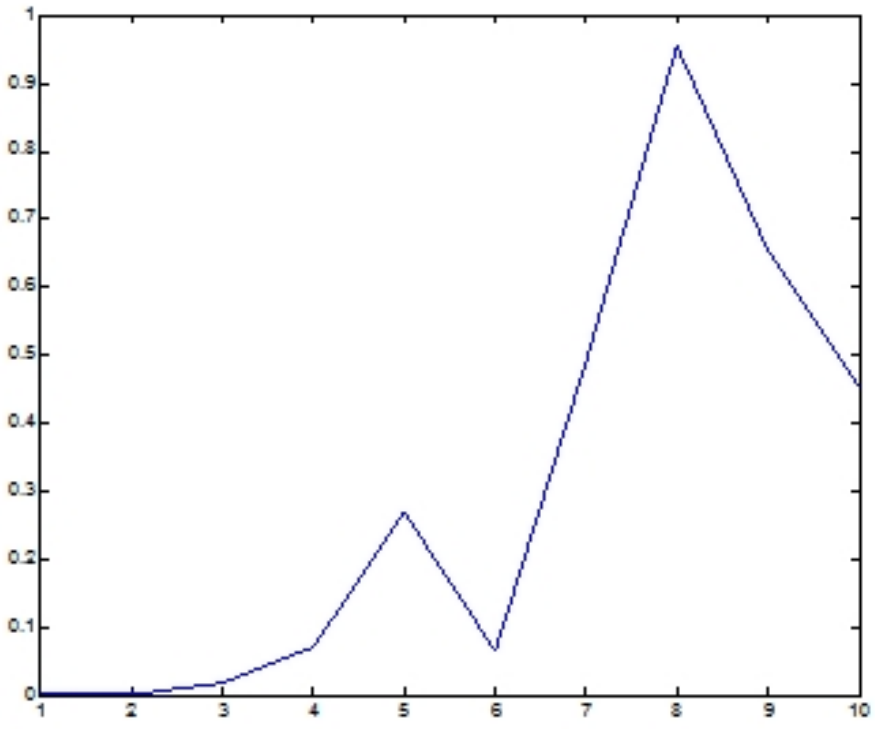

Figure 3.6 (Dataset 3) Absolute value of inner product of the new target vector with $\underline{w_{ie}}$ versus i, the index of the Eigenvectors. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

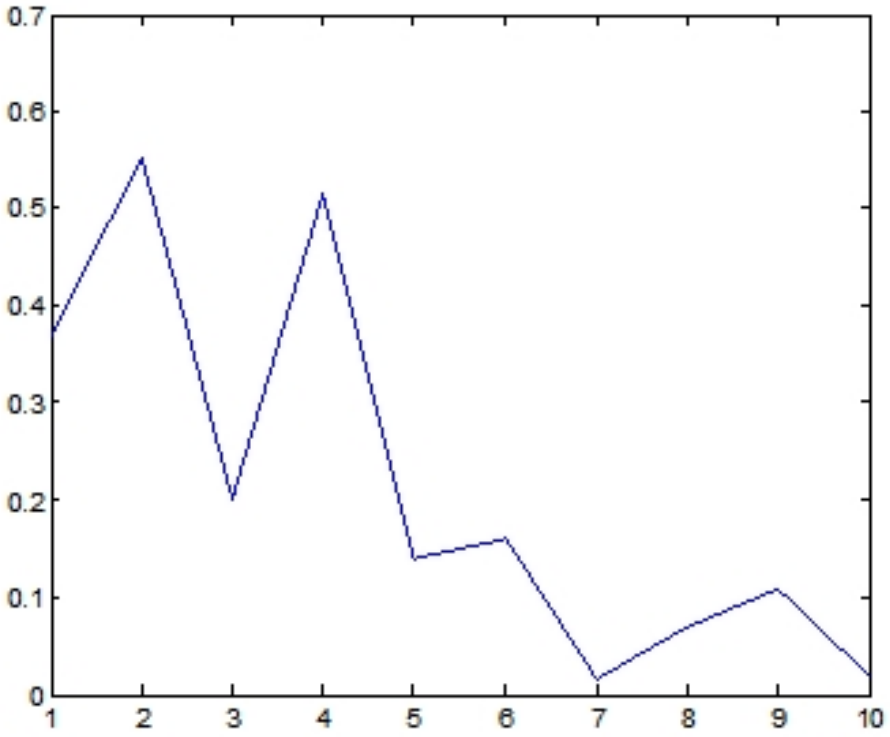

Figure 3.7 (Dataset 3) Absolute value of inner product of a clutter vector with $\underline{\hat{w}_i}$ versus i, the index of the $\underline{\hat{w}_i}$'s. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

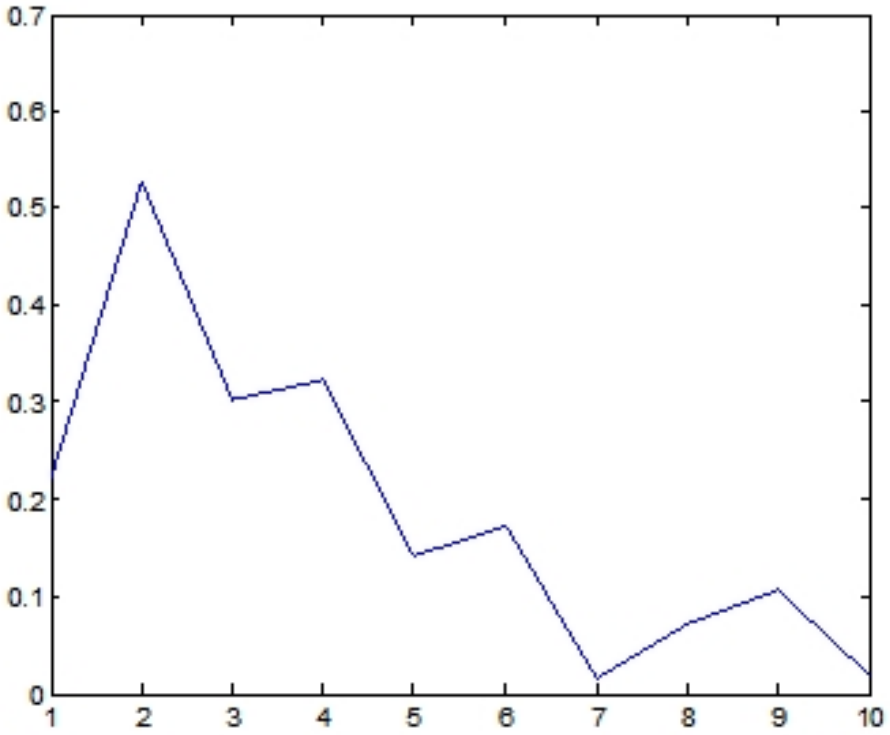

Figure 3.8 (Dataset 3) Absolute value of inner product of a clutter vector with $\underline{\hat{w}_{ie}}$ versus i, the index of the $\underline{\hat{w}_{ie}}$ s. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

OAEVD - Simulation Results for Infrared (IR) Data

The adaptive algorithm was also tested using some Infrared (IR) data provided by Lockheed Martin, MFC. This data consists of an Infrared (IR) video sequence (388 frames, each of size 126x126) of a target (tank) as a camera approaches it. Sample frames are shown in Figures 3.9 & 3.10.

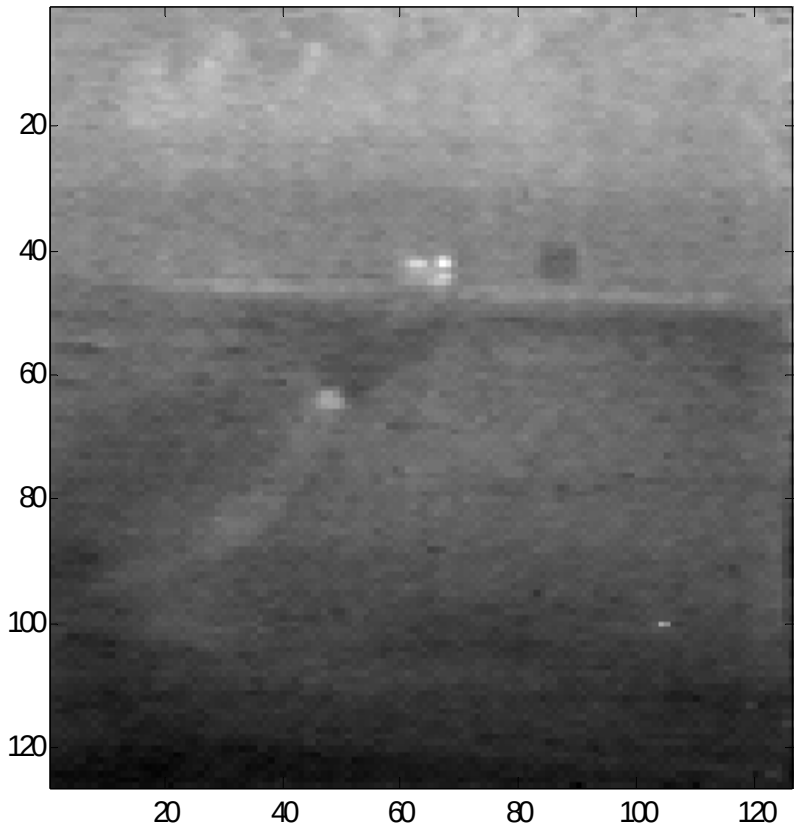

Figure 3.9 Sample frame 1

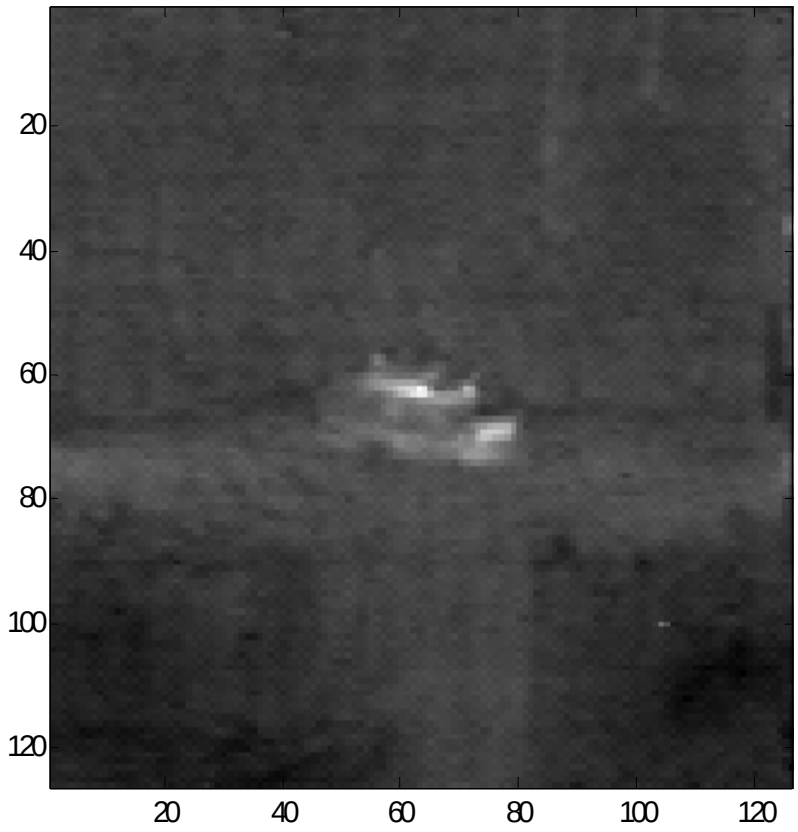

Figure 3.10 Sample frame 2

The idea is to use a certain number of frames as data for the training set, and the remaining frames as the new data. 16x16 chips of target and clutter are extracted from these frames. We obtain 409 target chips and 16,541 clutter chips. The reason for having a much larger number of clutter chips is because we have the freedom of picking anything that is not a target, as clutter. Therefore in each frame, we can pick multiple clutter chips compared to a few (sometimes one) target chip. Sample target and clutter chips are given in Figures 3.11 & 3.12 respectively.

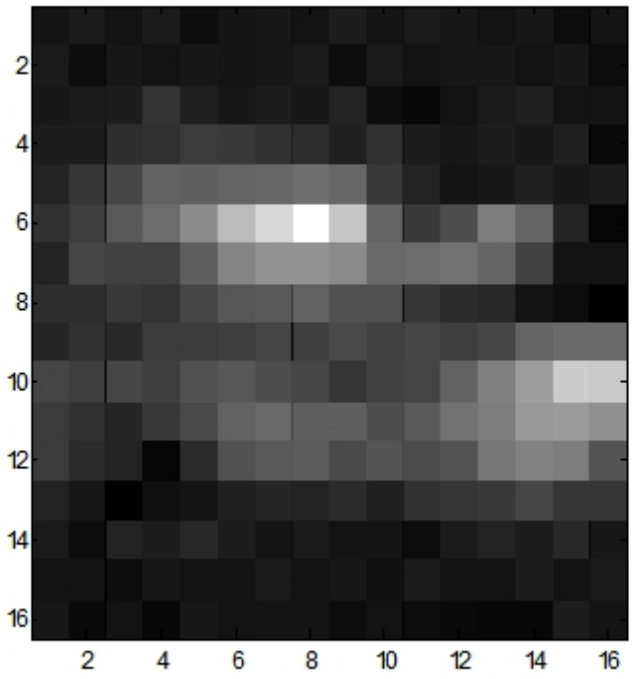

Figure 3.11 A sample target chip

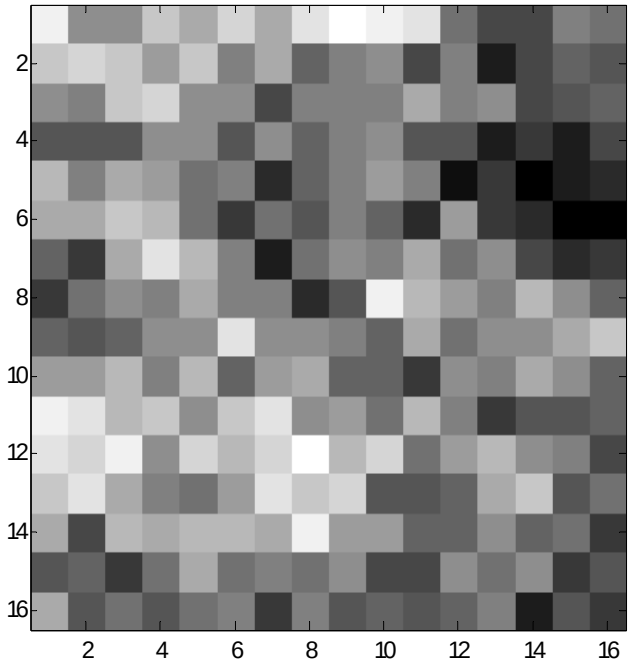

Figure 3.12 A sample clutter chip

Each chip is then converted into a 1D vector (256x1) by placing the columns one below the other. The target set and the clutter set are formed, by placing the respective chip vectors as columns of a matrix. Thus the dimensions of the target set and the clutter set are 256x409 and 256x16541 respectively.

Some sample results are presented below. The first 350 data points of the target and clutter sets are used to form the initial target and clutter autocorrelation matrices (R_x and R_y) respectively. The A matrix is computed according to (1), and the Eigenvalue Decomposition of A is performed to obtain the initial set of eigenvalues and eigenvectors. Then, new data point(s) is/are

added and the OAEVD is used to track the changes in the eigenvalues and eigenvectors. For the sake of evaluating the result, we compare the results of the OAEVD with the results obtained by performing the actual EVD on A_hat. In the simulation results, the following notation is used:

"lambda_orig" - initial eigenvalues

"W_orig" - initial eigenvectors

"lambda_hat_acc" – accurate eigenvalues

"W_hat_acc" - accurate eigenvectors

"lambda_hat" - eigenvalues obtained from the adaptive algorithm

"W_hat" – eigenvectors obtained from the adaptive algorithm

"iter" stands for the number of iterations.

Following the method in [88], we only track a few dominant eigenpairs, in this case 12. Adaptation is terminated when the change in the eigenvectors from one iteration to the next falls below a certain threshold epsilon. μ is the convergence factor as explained in the Formulation. All simulations are done using MATLAB 7.

Sample Result 1:

In Sample Result 1, one new data point (no. 351) is now incorporated into the target set, the new (perturbed) target autocorrelation matrix is obtained, and the A matrix is recalculated. The

OAEVD algorithm is now applied to calculate the new eigenvalues and eigenvectors corresponding to the modified A matrix.

Initial Target & Clutter sets – 350 points

Add 1 new target point – no. 351

Track 6 +ve and 6 –ve Eigenpairs, μ=0.001; epsilon=1e-16; Test point: No. 351

lambda_orig (initial eigenvalues) =

-0.9978 -0.9754 -0.9723 -0.9667 -0.9635 -0.9570 0.9923 0.9958 0.9965 0.9979 0.9989 0.9991

lambda_hat_acc (obtained by direct eigendecomposition) =

-0.9978 -0.9754 -0.9721 -0.9659 -0.9632 -0.9530 0.9924 0.9958 0.9965 0.9979 0.9989 0.9991

lambda_hat (obtained from the OAEVD algorithm) =

-0.9978 -0.9754 -0.9721 -0.9660 -0.9632 -0.9531 0.9924 0.9958 0.9965 0.9979 0.9989 0.9991

iter (number of iterations) =

2 8 25 11 13 14 3 3 2 26 2 3

Please note that the Eigenvectors (W_hat, W_hat_acc & W_hat) are not listed because the matrices are too large.

Sample Result 2:

In Sample Result 2, five new data points (no. 351-355) are now incorporated into the target set, the new (perturbed) target autocorrelation matrix is obtained, and the A matrix is recalculated. The OAEVD algorithm is now applied to calculate the new eigenvalues and eigenvectors corresponding to the modified A matrix.

Initial Target & Clutter sets – 350 points

Add 5 new target points – no. 351-355

Track 6 +ve and 6 –ve Eigenpairs, μ=0.001; epsilon=1e-16; Test point: No. 351

lambda_orig (initial eigenvalues) =

-0.9978 -0.9754 -0.9723 -0.9667 -0.9635 -0.9570 0.9923 0.9958 0.9965 0.9979 0.9989 0.9991

lambda_hat_acc (obtained by direct eigendecomposition) =

-0.9978 -0.9747 -0.9715 -0.9633 -0.9612 -0.9533 0.9927 0.9958 0.9966 0.9979 0.9989 0.9991

lambda_hat (obtained from the OAEVD algorithm) =

-0.9978 -0.9747 -0.9716 -0.9632 -0.9610 -0.9532 0.9927 0.9958 0.9966 0.9979 0.9989 0.9991

iter (number of iterations) =

12 19 13 21 13 17 6 2 5 4 2 21

Since the filter coefficients of the RQQCF are obtained from the Eigenvectors of A, it is expected that the absolute value of the inner product (response) of a target data point with each of the Eigenvectors of A, that correspond to the Eigenvalues close to +1, should be a large number compared to the absolute value of the inner product of a clutter data point with the same Eigenvectors. Conversely, the inner product of a clutter data point with each of the Eigenvectors corresponding to Eigenvalues close to -1, should be a large number compared to the inner product of a target point with the same Eigenvectors.

In Figures 3.13 – 3.18, the X-axis is the index of the Eigenvectors. The Y-axis is the absolute value of the inner product between a data point and the Eigenvectors. Also, the first six Eigenvectors correspond to clutter (eigenvalues close to -1) and the next six correspond to target (eigenvalues close to +1).

Figure 3.13 shows a plot of the absolute value of inner product of the 351st target vector (which was the new data point that was added) with W_hat & Figure 3.14 shows a plot of the absolute value of inner product of the same data point with W_hat_acc.

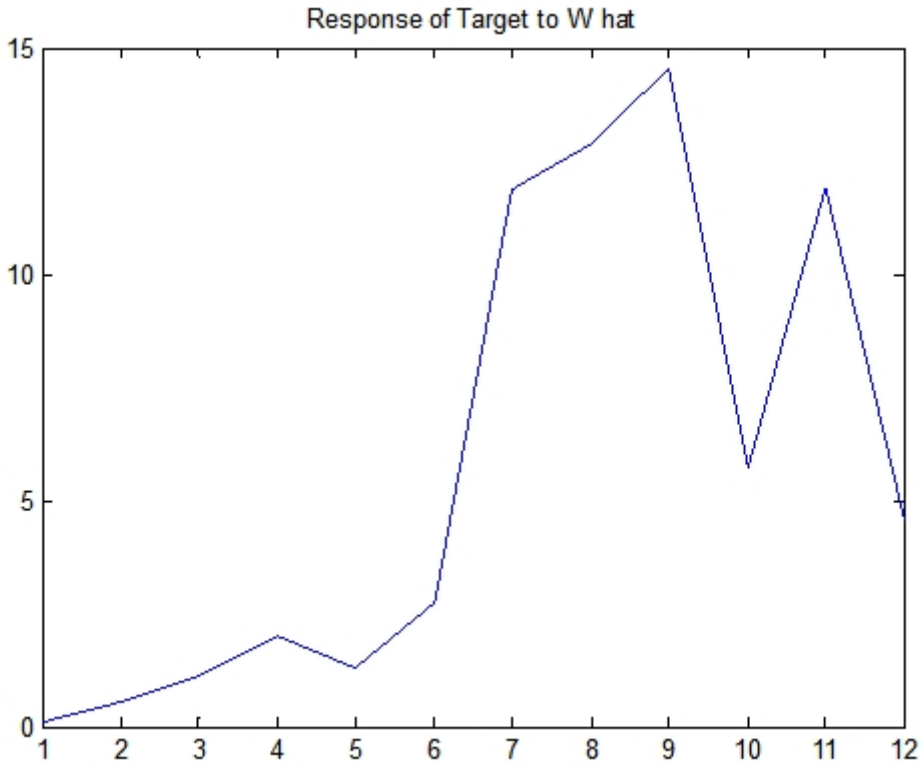

Figure 3.13 Absolute value of inner product of the 351st target vector (which was the new data point that was added) with W_hat

Figure 3.14 Absolute value of inner product of the same data point with W_hat_acc

Figure 3.15 shows the absolute value of inner product of the 200th clutter vector with W_hat & Figure 3.16 shows the absolute value of inner product of the same data point with W_hat_acc. Clutter data point 200 was chosen randomly just to illustrate how the response/inner product of a clutter data point with the Eigenvectors is different from that of a target data point.

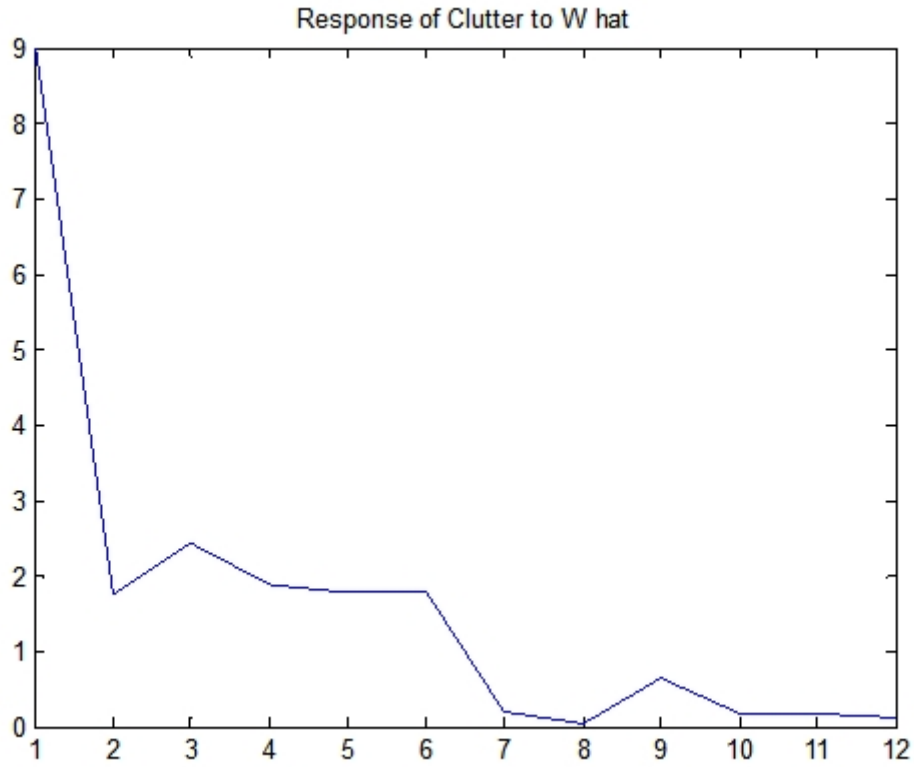

Figure 3.15 Absolute value of inner product of the 200th clutter vector with W_hat

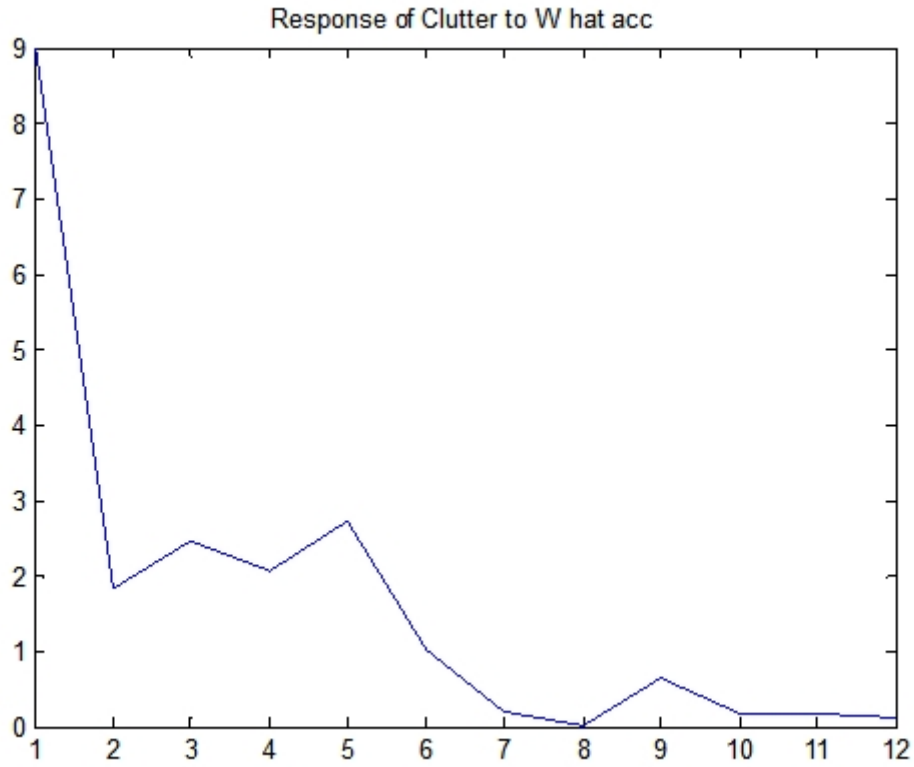

Figure 3.16 Absolute value of inner product of the same data point with W_hat_acc

As can be seen, the inner product of a target vector with eigenvectors 7 to 12 results in higher values overall than the inner product of a target vector with eigenvectors 1 to 6, and reciprocally, the inner product of a clutter vector with eigenvectors 1 to 6 results in higher values overall than the inner product of a clutter vector with eigenvectors 7 to 12.

In addition, there is a good match between the response using W_hat_acc (obtained from the direct Eigendecomposition of the updated A matrix) and W_hat (obtained from the OAEVD).

Figures 3.17 & 3.18 correspond to Sample Result 2. Figure 3.17 shows a plot of the absolute value of inner product of the 351st target vector (which was one the new data point that was added) with W_hat & Figure 3.18 shows a plot of the absolute value of inner product of the same data point with W_hat_acc.

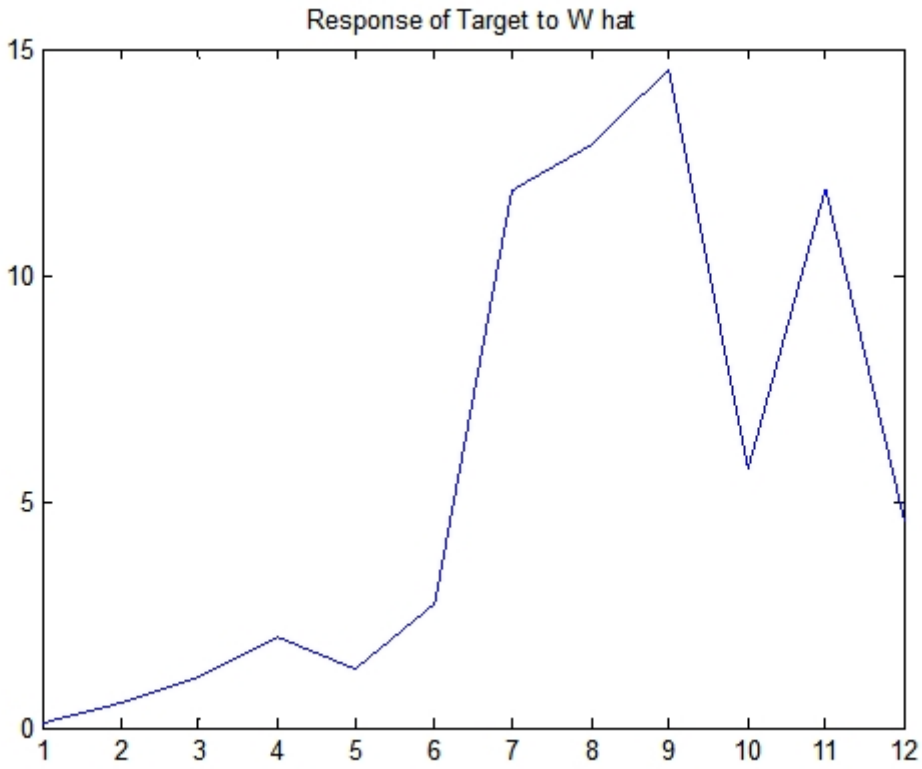

Figure 3.17 Absolute value of inner product of the 351st target vector (which was one the new data point that was added) with W_hat

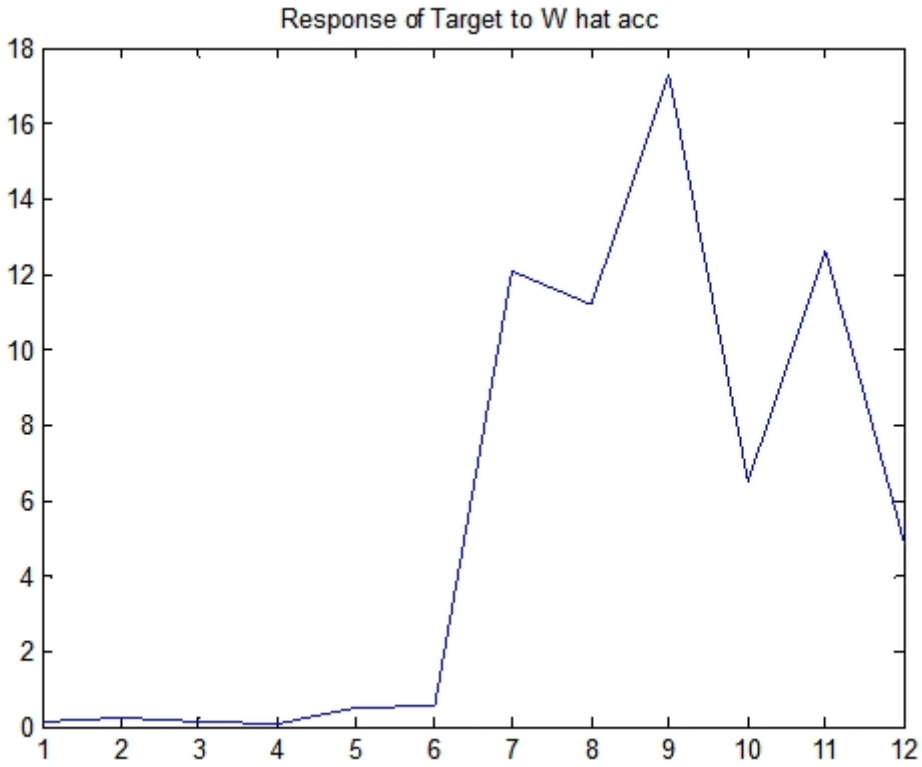

Figure 3.18 absolute value of inner product of the same data point with W_hat_acc

Summary

A novel algorithm for adaptive ATR based on the RQQCF technique was presented. When a few new data points have to be incorporated, the OAEVD algorithm provides substantial computational savings when tracking changes in the EVD. This is accomplished by using the old Eigenvalues and Eigenvectors to search for the new ones thus eliminating the need to perform matrix inversion, and direct EVD. The computational complexity of the Inversion and EVD

operations is of the order $O(n^3)$ for each operation, where n is the dimensionality of the correlation matrices. The computational complexity of the OAEVD is of the order $O(n^2k)$, where k is the number of eigenpairs to be tracked. The OAEVD algorithm can track any desired number of eigenpairs, although in practice only the dominant ones are needed. Additionally, the OAEVD algorithm can track them independent of each other, lending itself to parallel implementations. Sample results using synthetic and real IR datasets confirm the excellent properties of the OAEVD.

CHAPTER FOUR: THE TRANSFORM DOMAIN RQQCF (TDRQQCF)

As can be seen from the Chapter 2, the RQQCF technique operates on spatial domain data. Furthermore, each two-dimensional data chip in the spatial domain is converted into a one-dimensional vector by the lexicographical ordering of the columns of the chip. This leads to two interrelated issues. Firstly, the spatial structure in the two-dimensional chip is lost by converting it into a vector as described above. Secondly, the dimensionality of the system is increased considerably. One way to tackle both these issues simultaneously is to synthesize the RQQCF in the transform or frequency domain. Transforms capture the spatial correlation in images, and de-correlate the pixels. Consequently, if the transforms are appropriately selected, they compact the energy in the image in relatively few coefficients. Thus spatial domain data is transformed into an efficient and compact representation.

Transform Domain Processing

In the transform operation, a block of data is transformed into another representation which compacts the energy in the input, in a relatively few coefficients. These coefficients are further processed depending on the application, for e.g., in compression, these coefficients are encoded using quantization to achieve compression. The transform operation can be viewed as a method to obtain a sequence or a block of approximately uncorrelated coefficients from a highly correlated input sequence or block by removing the redundancy in the input signal. For images,

the transforms are applied to either the entire image or sub-images extracted from the image. Since an image or a sub-image is a two-dimensional block of pixels, image transforms are also two-dimensional. Generally, all two-dimensional (2D) transforms in use today are separable, i.e., the 2D transform is implemented as two one-dimensional transforms, each along a different dimension of the input block [148], [149]. A general 2D transform is given as,

$$[J] = [T]\,[I]\,[T]^{T} \tag{31}$$

where, [I] is a 2D array of pixel values, [T] is the matrix containing the basis vectors and [J] is the transformed matrix.

Transform Efficiency

As mentioned earlier, the application of a 2D transform to a block of pixels results in a reduction of correlation and therefore with an appropriate coding scheme for the transform coefficients, compression can be achieved. The selection of a transform for a particular application depends to a large extent on two things – complexity of processing and energy packing ability. An optimal transform in terms of complexity would obviously be one that is very simple to implement. An optimal transform, in an energy packing sense, would be the one that would pack the energy in the least number of coefficients.

Specific transforms for Images

Many transforms have been developed and reported in the literature in the context of image compression. Some of these are the Discrete Fourier Transform (DFT), the Discrete Cosine Transform (DCT), the Discrete Sine Transform (DST), the Discrete Hadamard Transform (DHT), the Karhunen-Loeve transform, also known as the Hotelling transform. It has been found that among these, the Karhunen-Loeve transform (KLT) is the most efficient in terms of energy compaction. It is optimum in the sense that it packs the most energy in the least number of coefficients, it minimizes the total entropy of the image and it completely decorrelates the pixels in the image [150]. In spite of this obvious advantage, the KLT has many implementation shortcomings, including the fact that its basis functions are dependent on the second order statistics and the size of the image. This makes it undesirable for image coding applications in general. Among the other transforms listed earlier, the DCT, [90], is found to be the closest to the KLT in terms of energy packing efficiency while being image independent. It is also a real transform unlike the complex DFT. The DCT is widely used since it has desirable properties as well as fast implementations. The DCT used in this work, is defined for an input image A and output image B, each of size M x N, as follows,

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \tag{32}$$

$0 \le p \le M-1,\ 0 \le q \le N-1,$

For $p = 0, \alpha_p = 1/\sqrt{M}$, and for $1 \le p \le M-1,\ \alpha_p = 2/\sqrt{M}$.

For $q = 0$, $\alpha_q = 1/\sqrt{N}$, and for $1 \le q \le N-1$, $\alpha_q = 2/\sqrt{N}$.

Figure 4.1 compares 1-d basis functions for various transforms for a signal of length eight. Figure 4.2 shows the 8x8 2-D basis functions of the DCT.

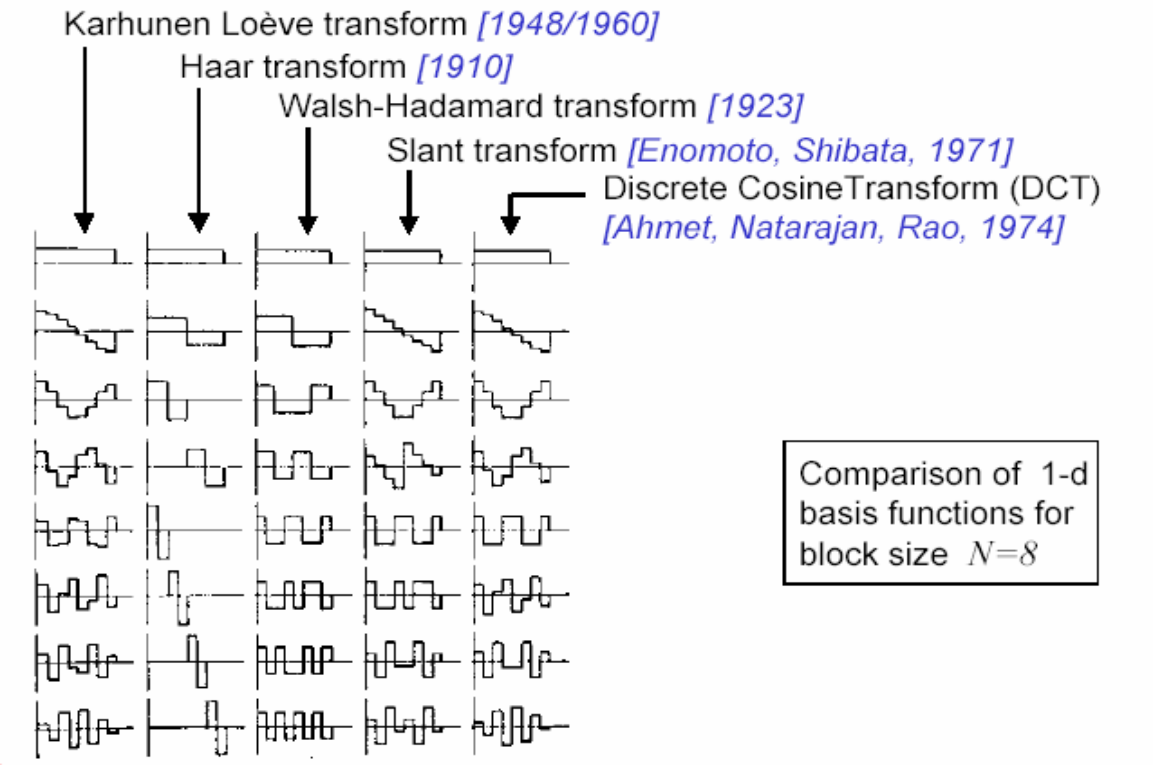

Figure 4.1 Comparison of 1-d basis functions for a signal of size N = 8 (from [149])

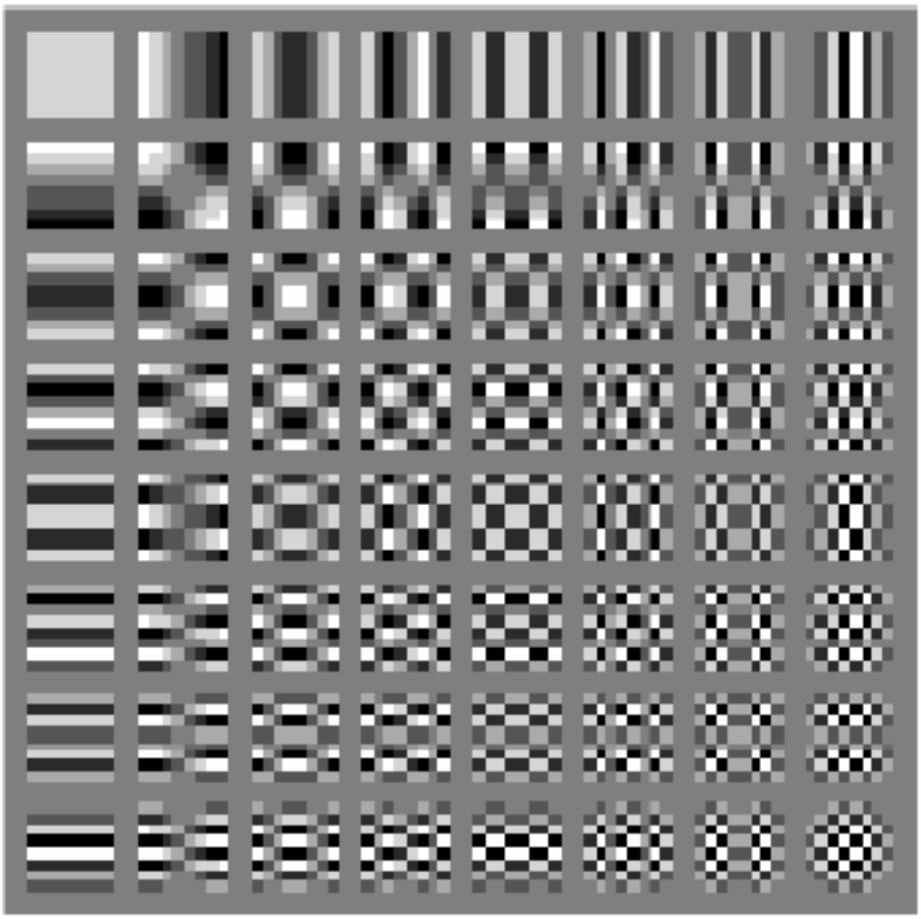

Figure 4.2 8x8 2-D basis functions of the DCT

The TDRQQCF Algorithm

The TDRQQCF technique proceeds as follows:

1. Each target chip, C_x, and clutter chip, C_y, is first transformed using the DCT to obtain C_{xt} and C_{yt}. It is seen that most of the energy in C_{xt} and C_{yt} is concentrated in the top left corner. In addition, the distribution of energy for targets and clutter differ from each other.

2. Each C_{xt} and C_{yt} is truncated to an appropriate size and converted to a one-dimensional vector by lexicographically ordering the columns. Thus, vectors of reduced dimensionality compared to the spatial domain case, are obtained. In addition, these vectors are very efficient representations of the spatial domain chips.

3. The autocorrelation matrices, R_{xt} and R_{yt} are computed, and used to obtain A_t according to Equation 5. We note that since the dimensions of the target and clutter vectors are much smaller than in the spatial domain case, the dimensionality of the autocorrelation matrices, R_{xt} and R_{yt}, and therefore A_t, are correspondingly reduced.

4. The Eigenvalue Decomposition (EVD) is performed on A_t to obtain the QCF coefficients. The QCF coefficients thus obtained are in the DCT domain.

Figure 4.3 shows the steps of the TDRQQCF.

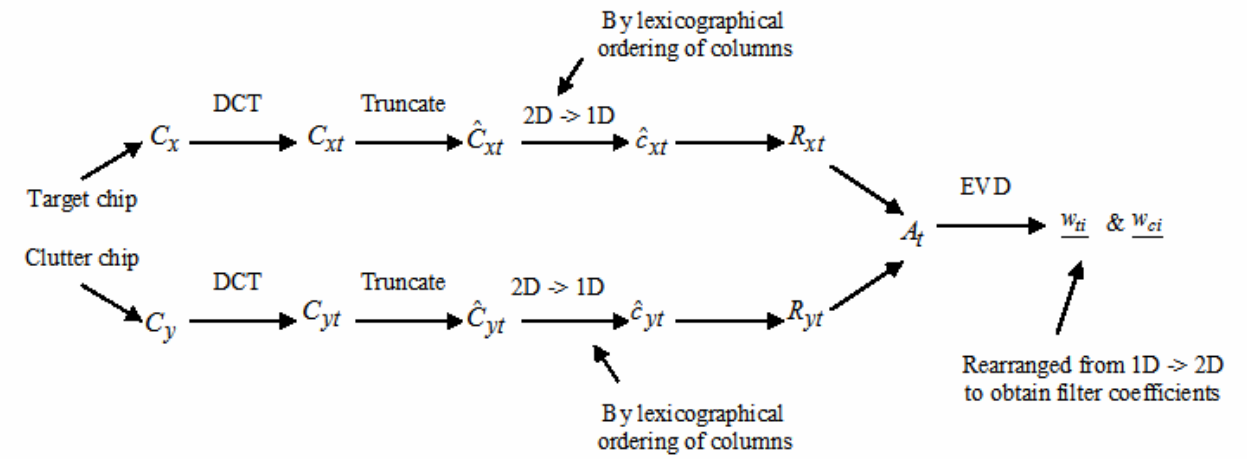

Figure 4.3 Steps of the TDRQQQCF

Figure 4.4(a) shows a 16x16 target chip in the spatial domain. Figure 4.4(b) shows the same chip after transformation to the DCT domain. Note the concentration of the energy in the chip in a very small number of coefficients in the DCT domain.

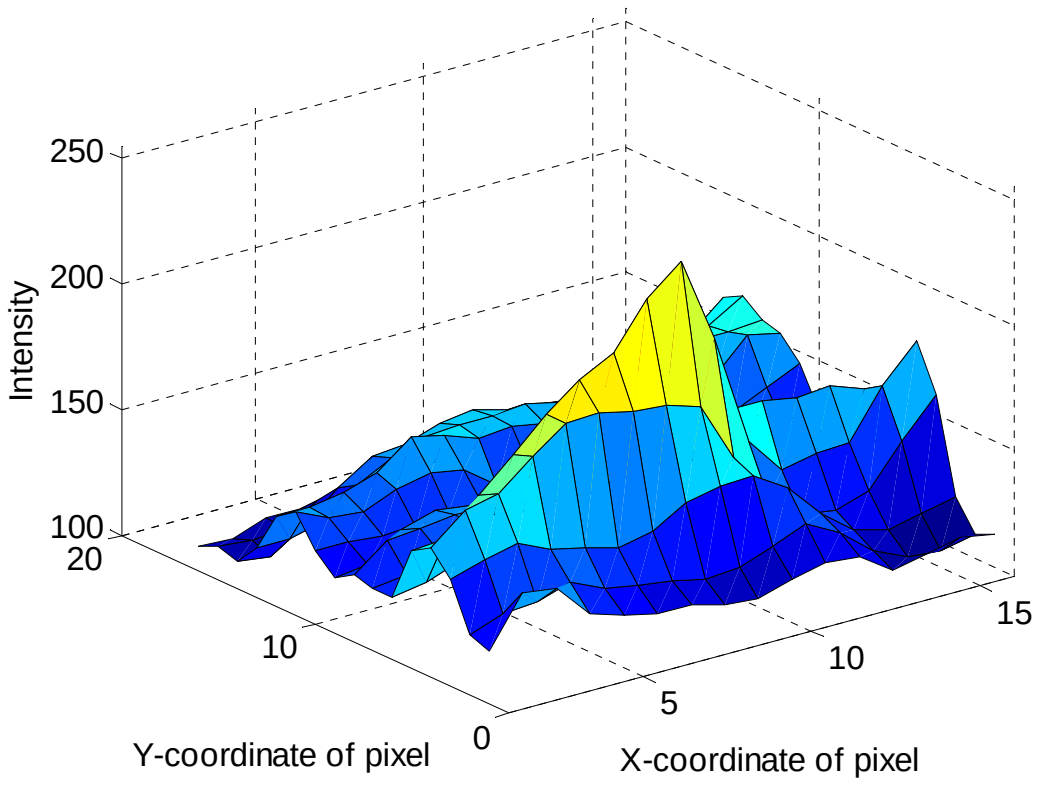

Figure 4.4(a) A sample target chip in the spatial domain

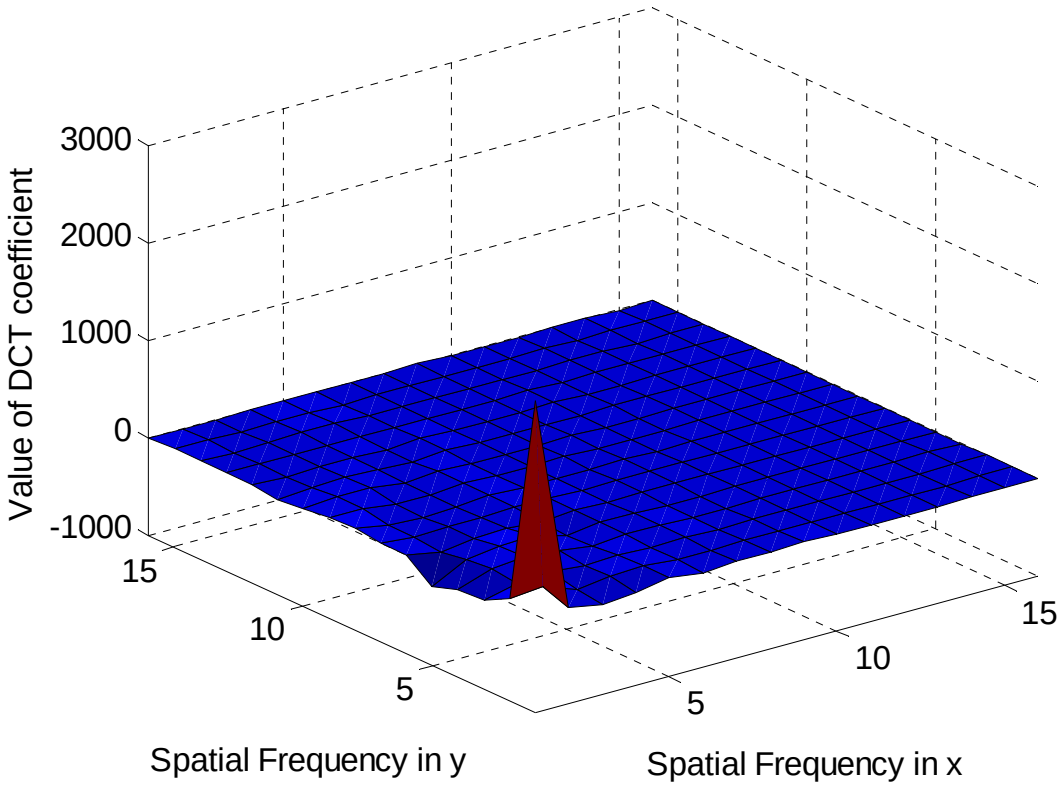

Figure 4.4(b) Target chip corresponding to previous figure in the DCT domain

Simulation Results

The TDRQQCF was tested on various Infrared video sequences provided by Lockheed Martin, Missile and Fire Control, (LMMFC). Sample results are presented from four video sequences to illustrate the performance of the proposed technique. Figure 4.5a - 4.5d show sample frames from the different videos.

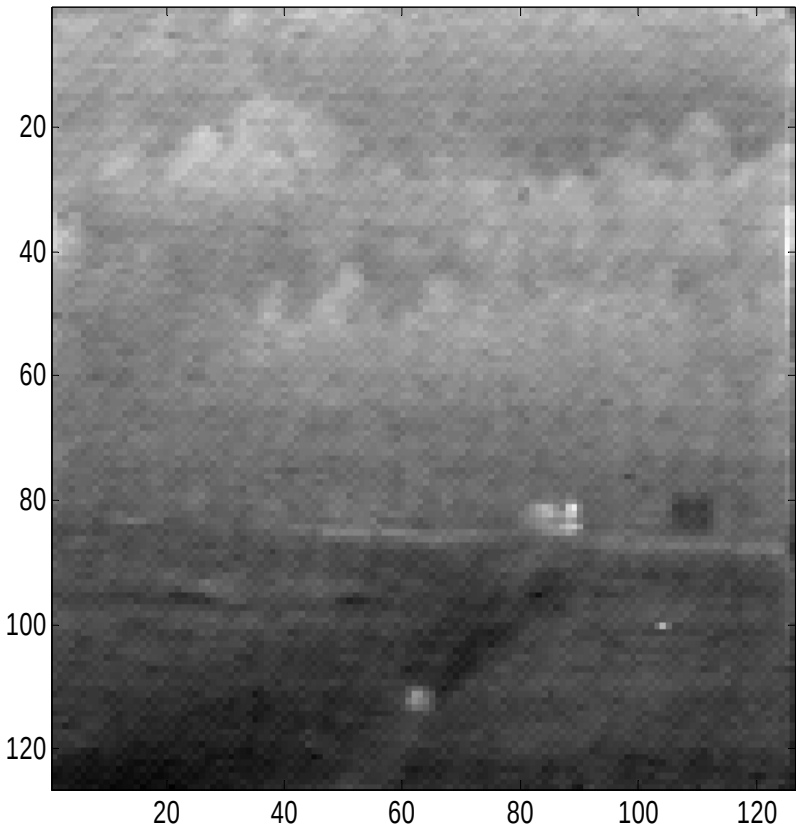

Figure 4.5(a) Sample frame from Video 1

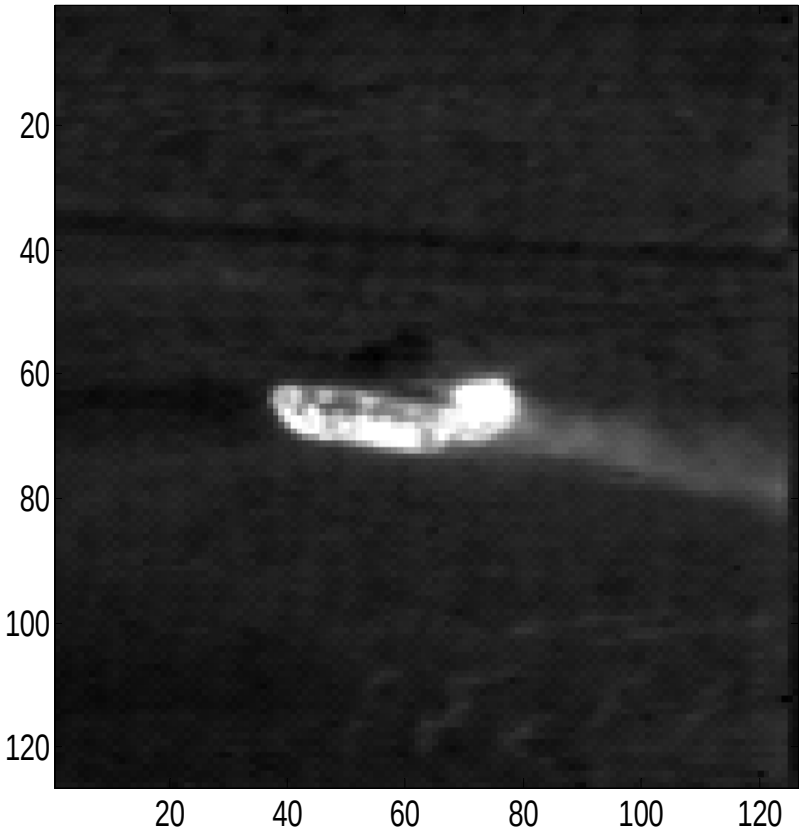

Figure 4.5(b) Sample frame from Video 2

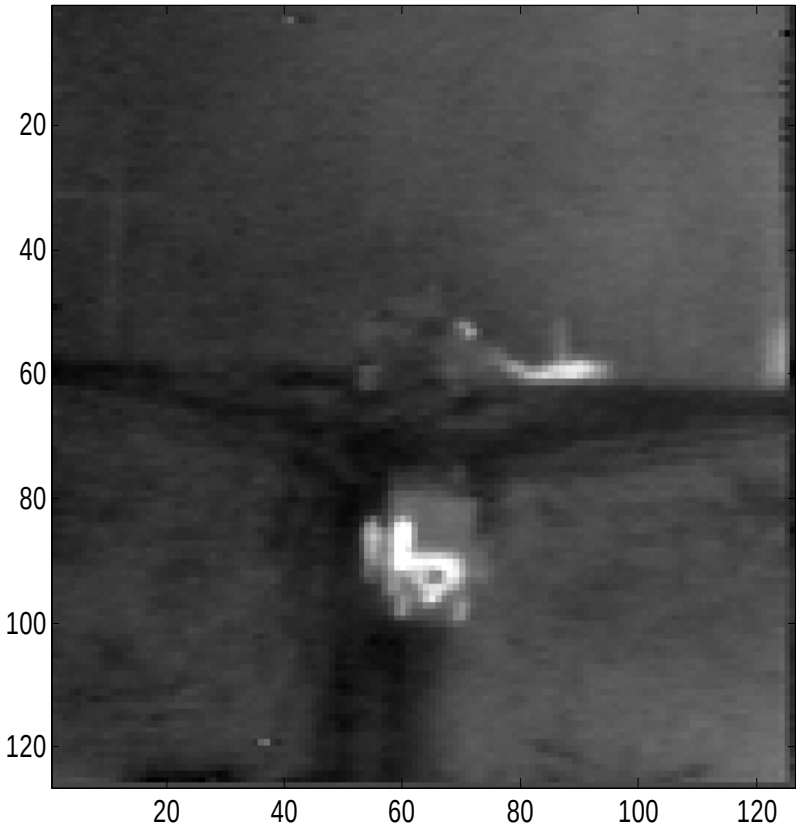

Figure 4.5(c) Sample frame from Video 3

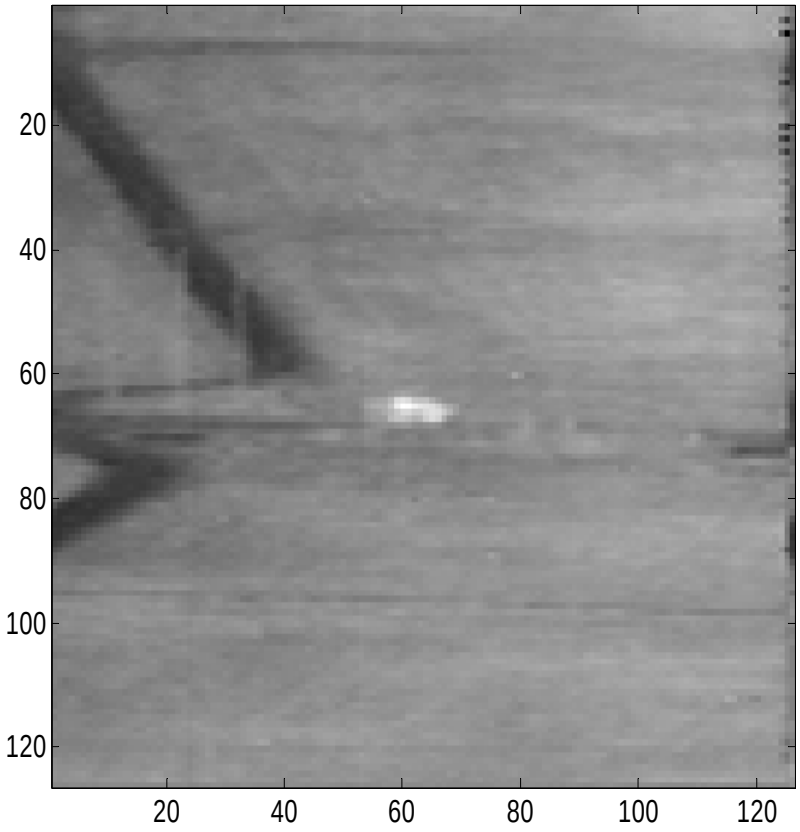

Figure 4.5(d) Sample frame from Video 4

Table 4.1 shows the number of frames in each video and the number of target and clutter chips, M, obtained for each video. Target chips are obtained from each frame of a video using ground truth data that is available. For clutter, chips are picked from all areas of each frame of the video except the area(s) where the target(s) is/are located. While this results in a larger number of clutter chips than target chips, for our simulations, the number of clutter chips is chosen to be equal to the number of target chips for convenience sake. Note that the size of the autocorrelation matrices depends only on the dimension of the data points and not on the number of data points.

Table 4.1 Number of Frames and Number of Target and Clutter chips, M, for each video.

	VIDEO 1	VIDEO 2	VIDEO 3	VIDEO 4
Number of Frames	388	778	410	300
Number of Target and Clutter Chips, M	409	763	405	391

The size of each chip is 16x16, i.e., $\sqrt{n}$ =16 and n=256. Figure 4.6 illustrates the advantage of transforming a target chip from VIDEO 1, using the 2D DCT before converting the transformed chip into a 1D vector, Figure 4.6b, versus converting the chip into a 1D vector first and then applying the 1D DCT, Figure 4.6a.

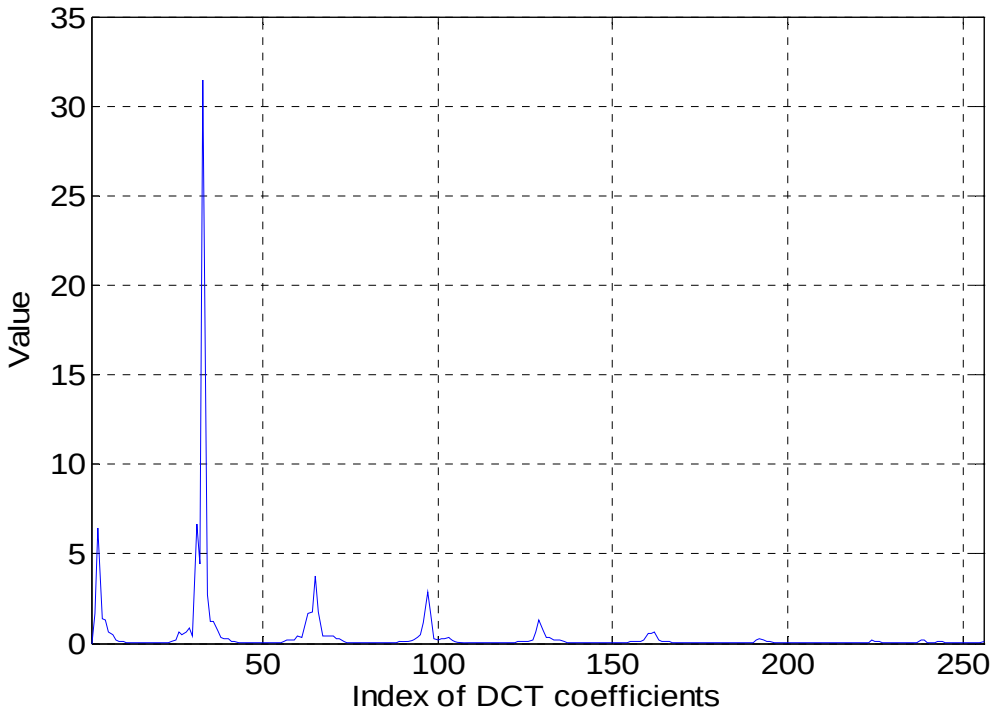

Figure 4.6(a) DCT coefficients obtained by converting a 2D target chip into a 1D vector before applying the 1D DCT

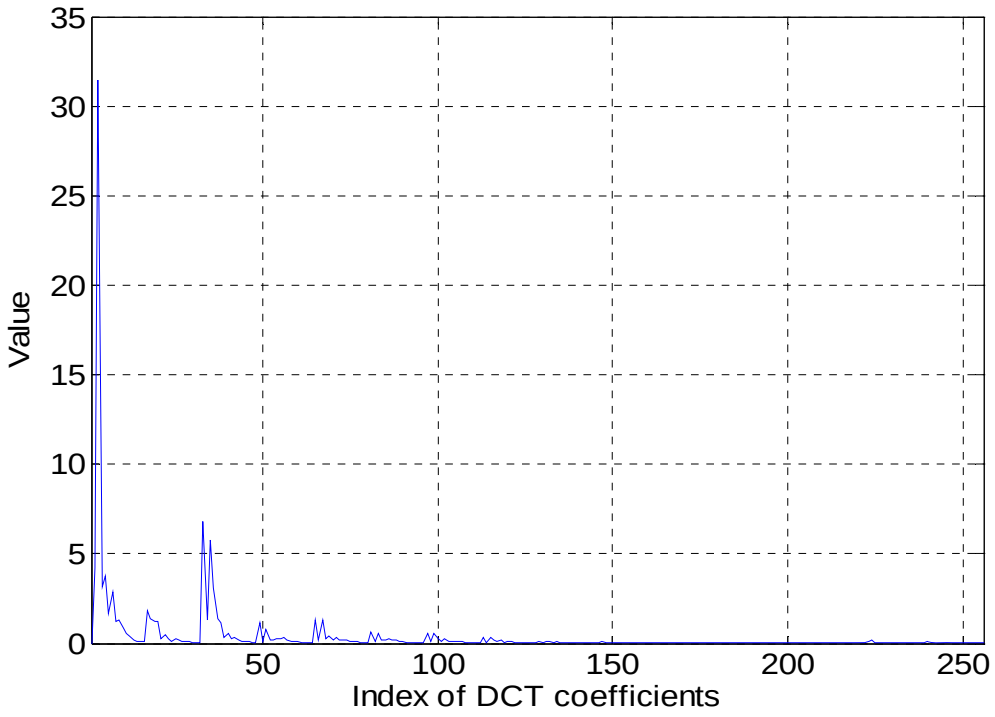

Figure 4.6(b) DCT coefficients obtained by first transforming the chip using the 2D DCT and then converting it to a 1D vector

It can be seen that first transforming the 2D chip results in better energy compaction, leading to efficient representation. Figure 4.7a and 4.7b show corresponding plots for a clutter point from VIDEO 1.

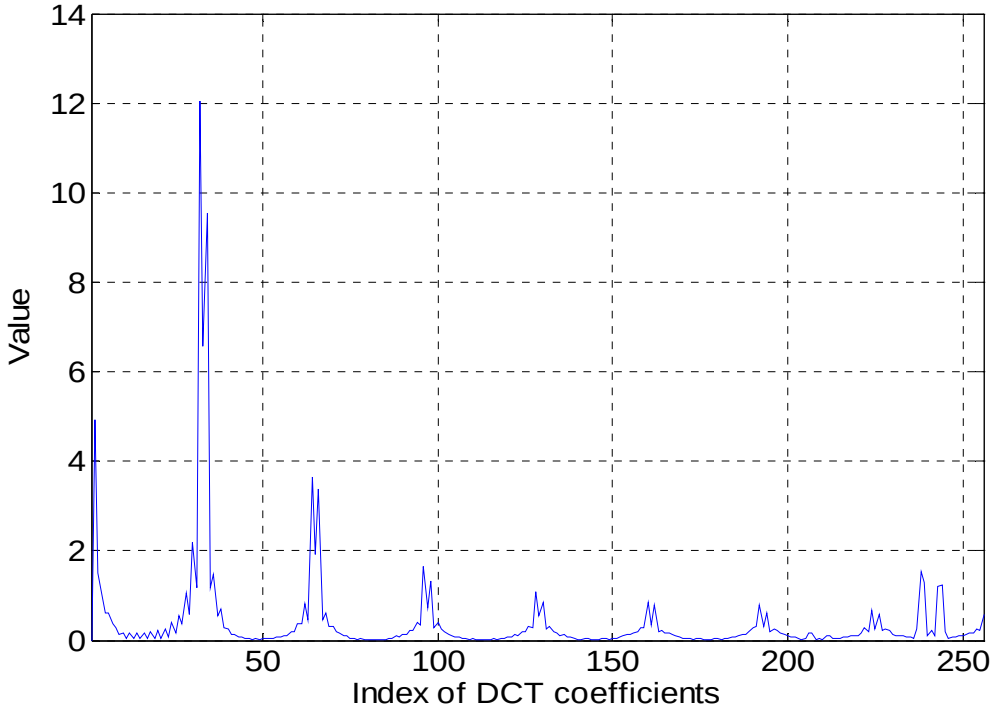

Figure 4.7(a) VIDEO 1: (a) DCT coefficients obtained by converting a 2D clutter chip into a 1D vector before applying the 1D DCT

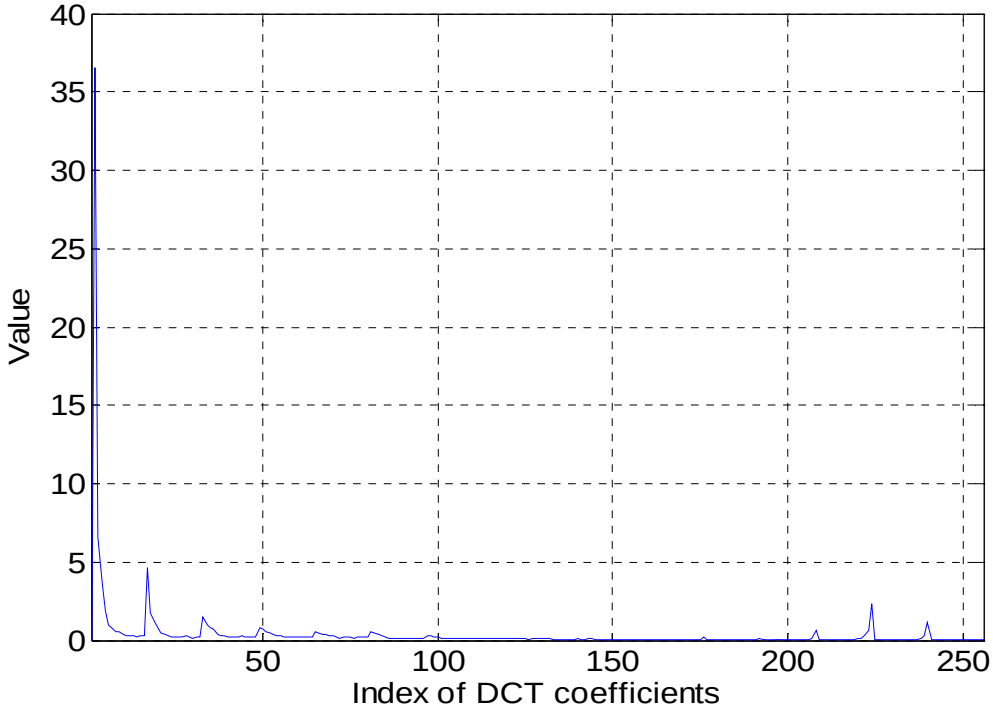

Figure 4.7(b) VIDEO 1: DCT coefficients obtained by first transforming the chip using the 2D DCT and then converting to a 1D vector

Tables 4.2 to 4.5 show, for IR Videos 1 to 4, respectively, the average energy in sub-images of different sizes, retaining the low spatial frequency region, of the transformed 16x16 chips. From these tables, 75% to 90% of the energy is concentrated in 25% of the transformed chips. Also, the target energy is slightly more compressed in the transform domain. In addition, the energy distribution for target chips is different from that for clutter chips.

Table 4.2 VIDEO 1: Avg. energy in different transformed and truncated matrices of the target and clutter sets.

Avg. Energy in →	8x8	9x9	10x10	11x11	12x12	13x13	14x14	15x15	16x16
Target chips	88.5799	91.8736	94.2695	95.8867	97.036	97.8537	98.5065	99.1296	100
Clutter chips	77.0727	79.5609	81.8025	83.8621	85.9592	87.9347	90.4363	93.6521	100

Table 4.3 VIDEO 2: Avg. energy in different transformed and truncated matrices of the target and clutter sets.

Avg. Energy in →	8x8	9x9	10x10	11x11	12x12	13x13	14x14	15x15	16x16
Target chips	95.3762	96.409	97.35	97.9219	98.4602	98.8663	99.2888	99.5657	100
Clutter chips	87.2741	89.2236	90.841	92.2614	93.6029	94.8183	96.5067	98.578	100

Table 4.4 VIDEO 3: Avg. energy in different transformed and truncated matrices of the target and clutter sets.

Avg. Energy in →	8x8	9x9	10x10	11x11	12x12	13x13	14x14	15x15	16x16
Target chips	93.86	95.4257	96.5763	97.443	98.1287	98.6852	99.1279	99.5433	100
Clutter chips	80.3775	83.1199	85.6899	88.1777	90.6858	93.0544	95.2862	97.7443	100

Table 4.5 VIDEO 4: Avg. energy in different transformed and truncated matrices of the target and clutter sets.

Avg. Energy in →	8x8	9x9	10x10	11x11	12x12	13x13	14x14	15x15	16x16
Target chips	94.4088	96.195	97.6136	98.3129	98.8124	99.1503	99.4638	99.649	100
Clutter chips	87.7344	89.888	91.8235	93.5835	95.1314	96.409	97.5693	98.7895	100

At this point, if no truncation is performed, and the original RQQCF technique is applied, the same set of eigenvalues as in the case of the spatial domain RQQCF is obtained. For example, for Video 1, twelve dominant eigenvalues (six positive and six negative) for both cases are listed below, (k = 12).

λ_i (Spatial domain)

-0.9975 -0.9641 -0.9559 -0.9378 -0.9283 -0.9221 0.9934 0.9938 0.9962 0.9971 0.9981 0.9985

λ_i (DCT domain)

-0.9975 -0.9641 -0.9559 -0.9378 -0.9283 -0.9221 0.9934 0.9938 0.9962 0.9971 0.9981 0.9985

Figure 4.8(a) shows the eigenvalue distribution for the spatial domain RQQCF technique while Figure 4.8(b) shows the eigenvalue distribution for the TDRQQCF technique when the chips are compressed from 16x16 to 8x8.

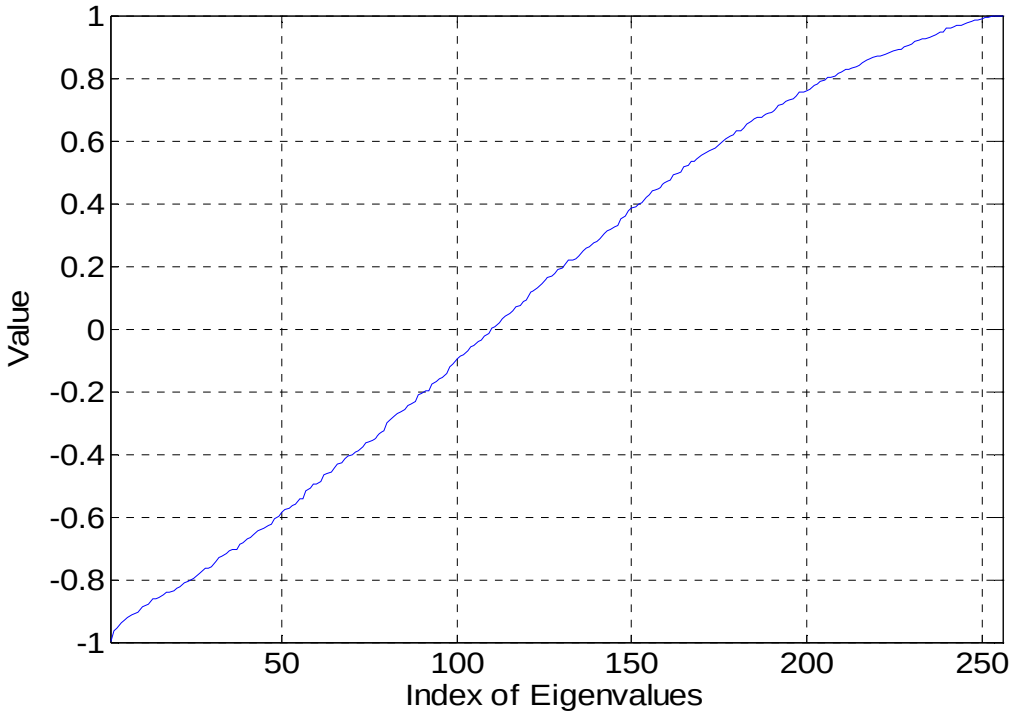

Figure 4.8(a) Distribution of eigenvalues in the spatial domain RQQCF method

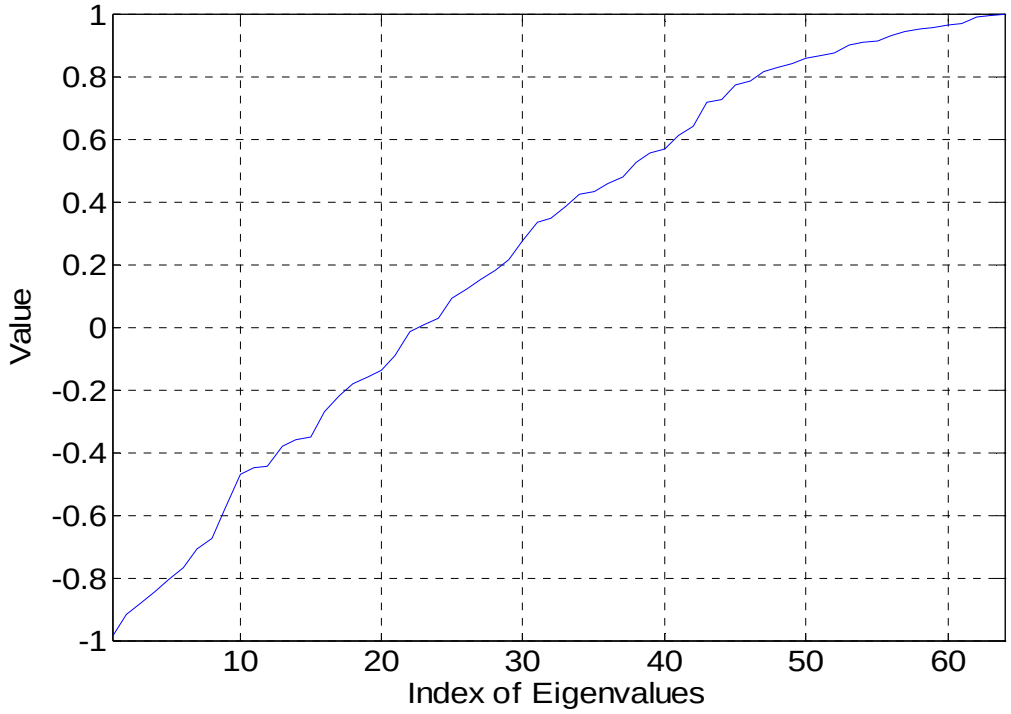

Figure 4.8(b) Distribution of eigenvalues in the TDRQQCF method for chips compressed to 8x8

Sample results are presented for the case when the transformed target and clutter chips, C_{xt} and C_{yt} respectively, are truncated to a 8x8 size. This means that $\sqrt{n}$ =8 and n=64. The twelve dominant eigenvalues among the 64 are listed below, (k = 12).

λ_i (DCT domain)

-0.9930 -0.8648 -0.7825 -0.7642 -0.7472 -0.6427 0.9642 0.9722 0.9746 0.9878 0.9943 0.9952

As explained earlier, to identify a data point as target or clutter, the sum of the absolute values of the k inner products of a data point with w_{ti} and w_{ci}, p_t and p_c, are calculated. If $p_t > p_c$, the data point is identified as a target. Otherwise, it is identified as clutter. We will refer to the absolute values of these inner products as the 'response' of that particular data point. Figures 4.9 – 4.16 show the response of representative target and clutter vectors plotted against the index of the dominant eigenvectors, in both the spatial and the DCT domains. Eigenvectors 1 – 6 are dominant eigenvectors for clutter while eigenvectors 7 – 12 are dominant eigenvectors for targets. Figures 4.9 and 4.10 correspond to VIDEO 1, Figures 4.11 and 4.12 correspond to VIDEO 2, Figures 4.13 and 4.14 correspond to VIDEO 3, and Figures 4.15 and 4.16 correspond to VIDEO 4. To explain further, Figure 4.9 shows for VIDEO 1, the absolute value of the inner product of representative target and clutter vectors with the eigenvectors corresponding to the original dominant eigenvalues (spatial domain), respectively, versus the index of the eigenvectors. Figure 4.10 shows for VIDEO 1, the absolute value of the inner product of the

same target and clutter vectors with the eigenvectors corresponding to the dominant eigenvalues obtained in the DCT domain, respectively, versus the index of the eigenvectors. This is repeated for the other three videos in figures 4.11 – 4.16.

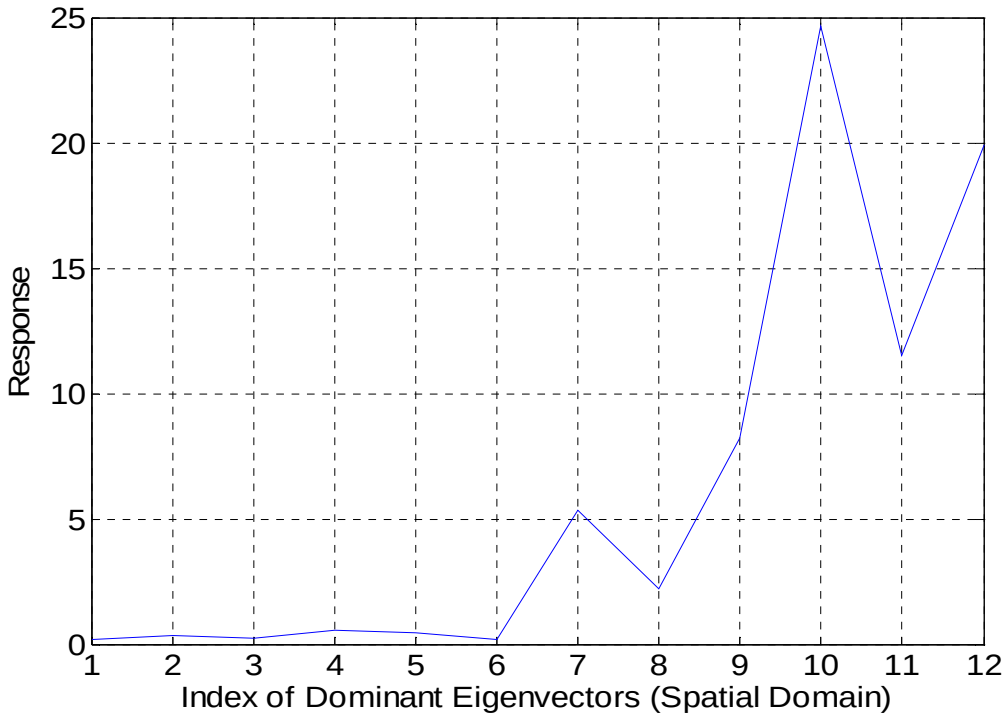

Figure 4.9(a) VIDEO 1: Response of (a) a representative target vector versus the index of the dominant eigenvectors (spatial domain)

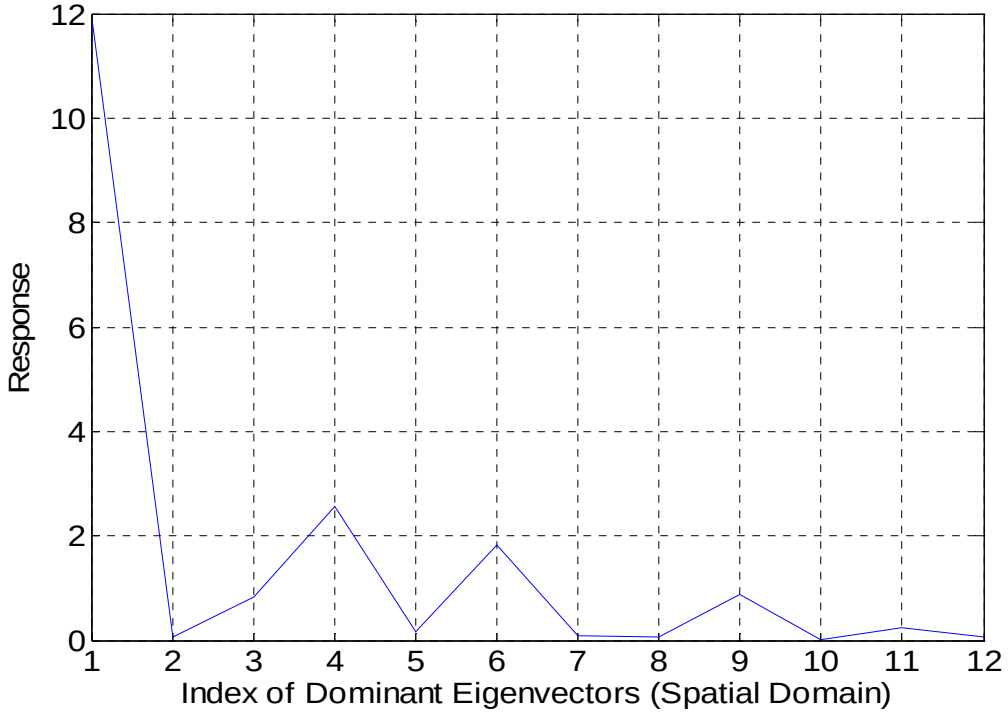

Figure 4.9(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors (spatial domain)

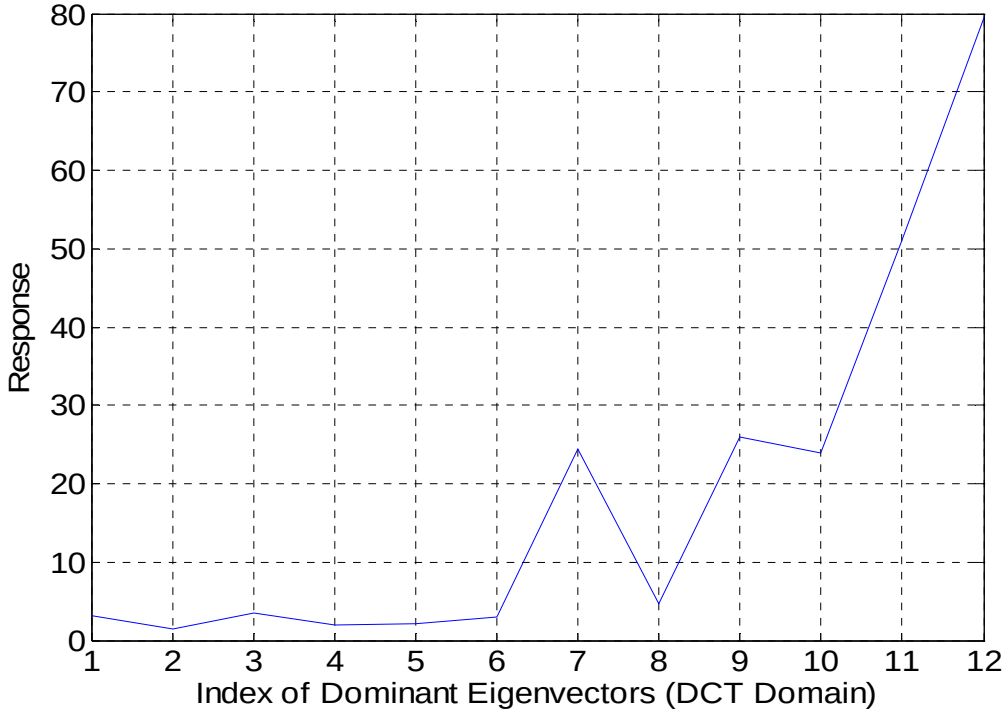

Figure 4.10(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

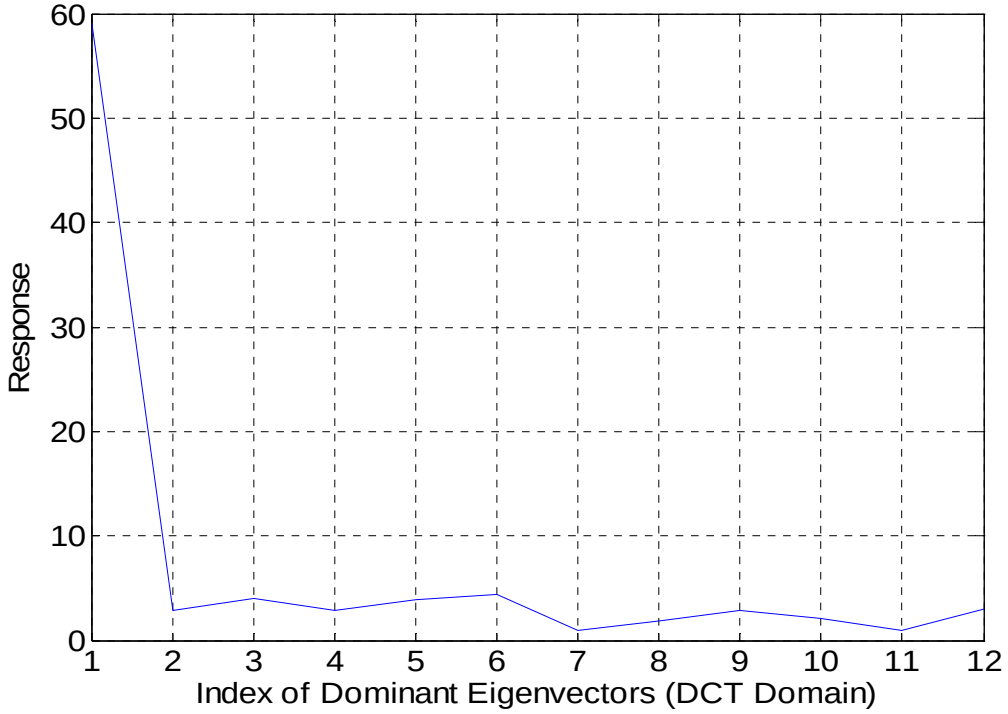

Figure 4.10(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

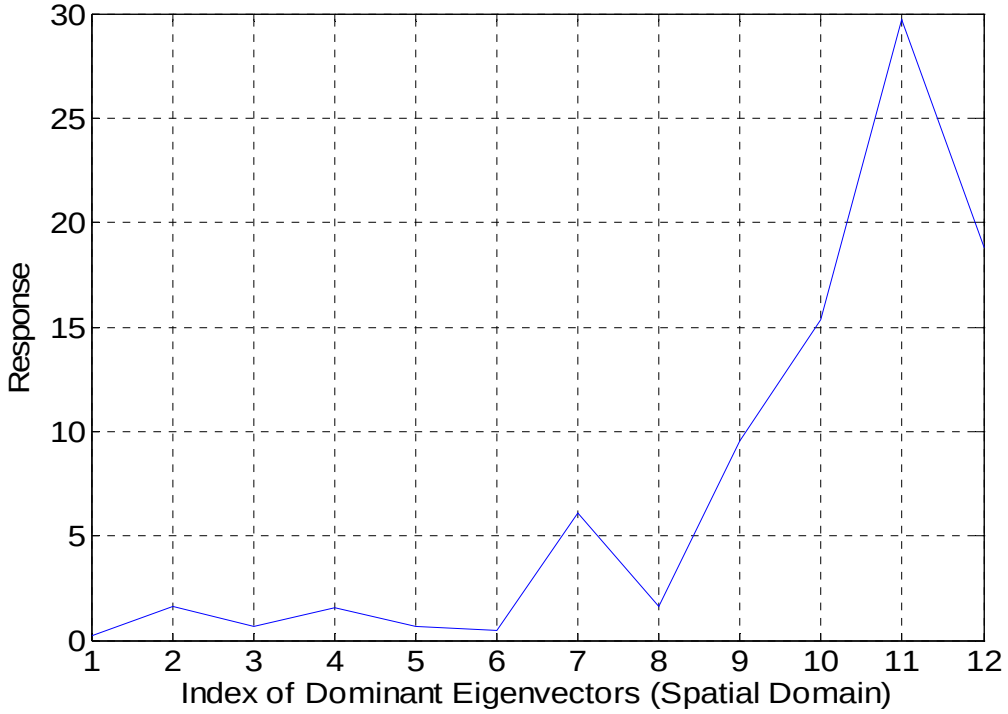

Figure 4.11(a) VIDEO 2: Response of (a) a representative target vector versus the index of the dominant eigenvectors (spatial domain)

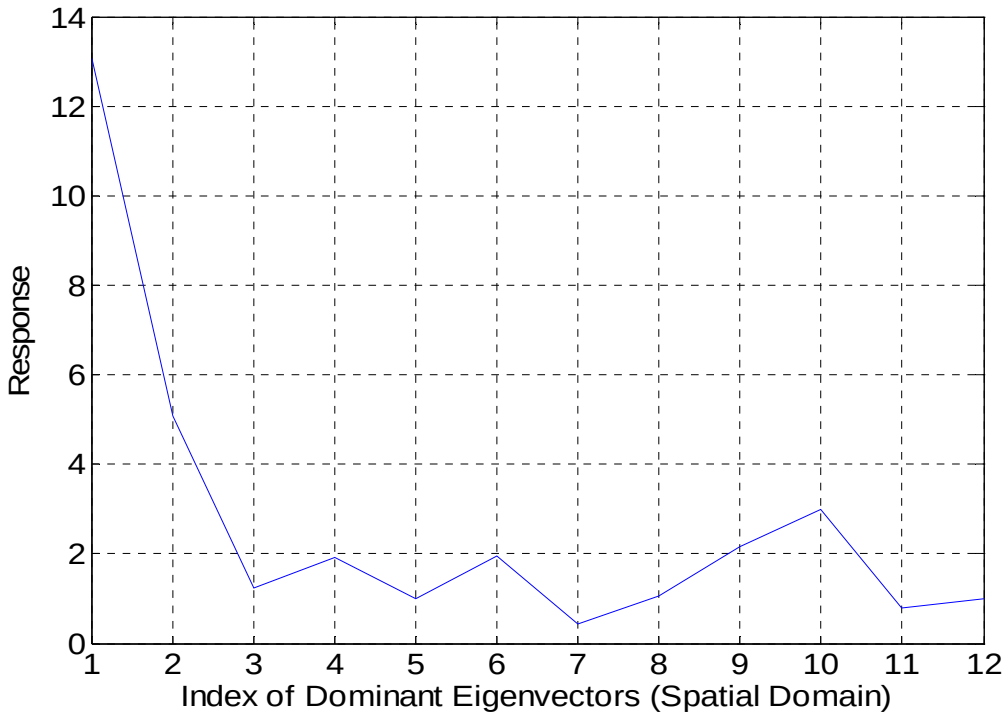

Figure 4.11(b) VIDEO 2: Response of a representative clutter vector versus the index of the dominant eigenvectors (spatial domain)

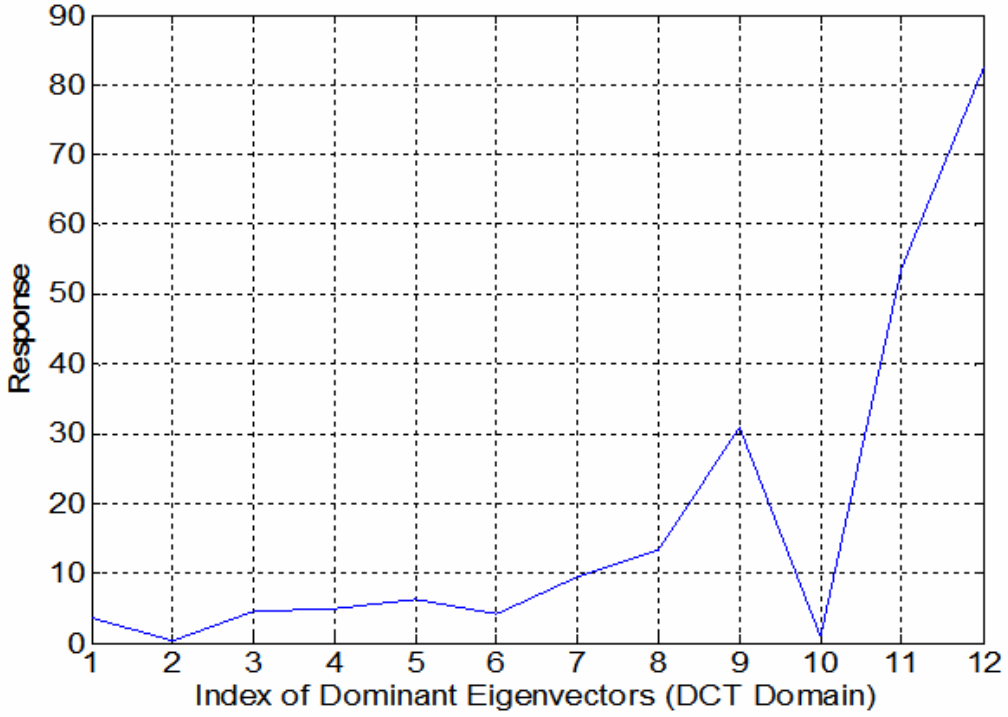

Figure 4.12(a) VIDEO 2: Response of a representative target vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

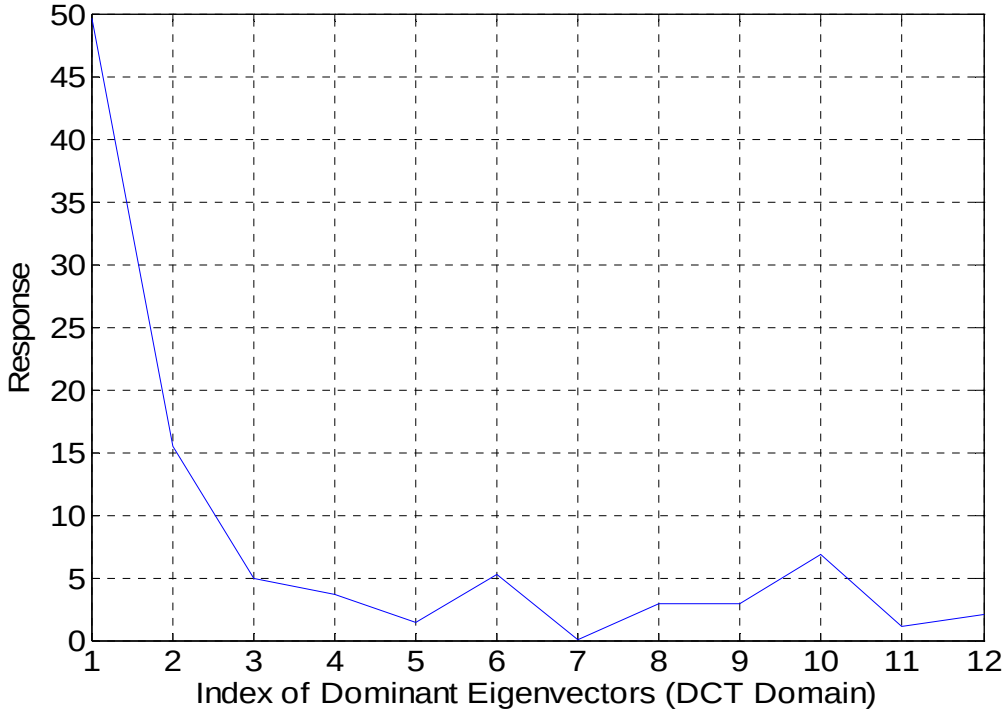

Figure 4.12(b) VIDEO 2: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

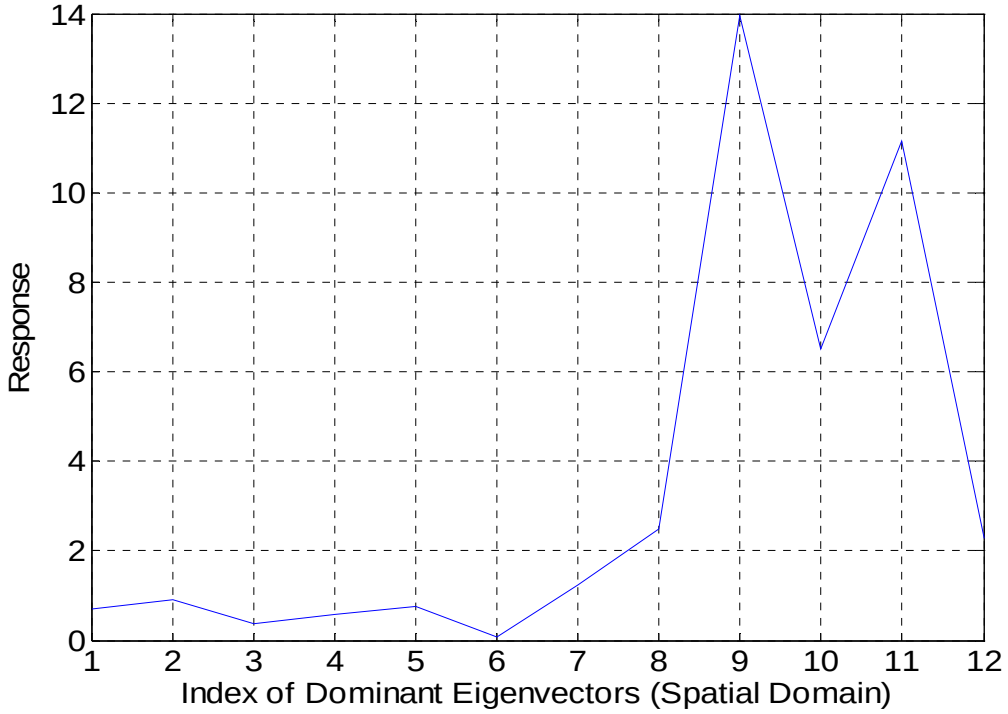

Figure 4.13(a) VIDEO 3: Response of (a) a representative target vector versus the index of the dominant eigenvectors (spatial domain)

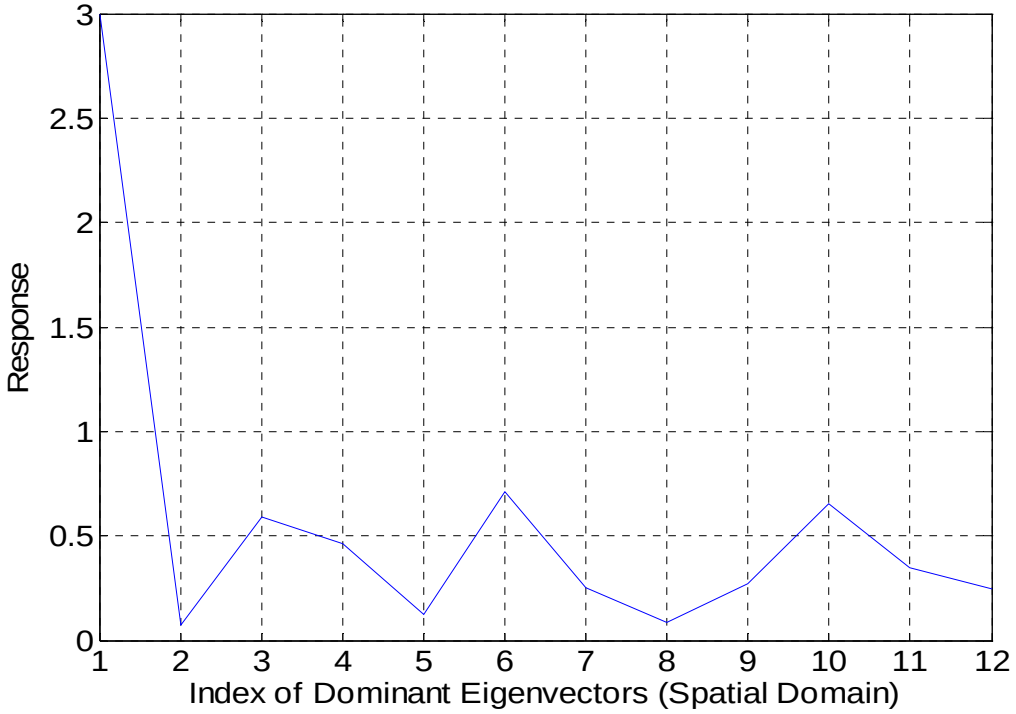

Figure 4.13(b) VIDEO 3: Response of a representative clutter vector versus the index of the dominant eigenvectors (spatial domain)

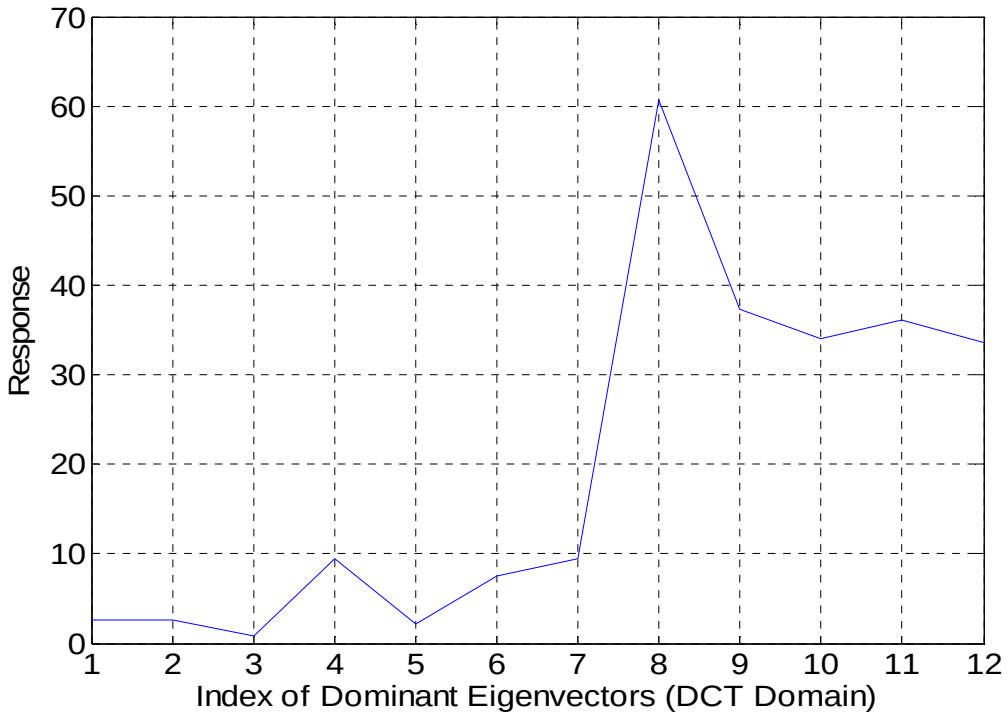

Figure 4.14(a) VIDEO 2: Response of a representative target vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

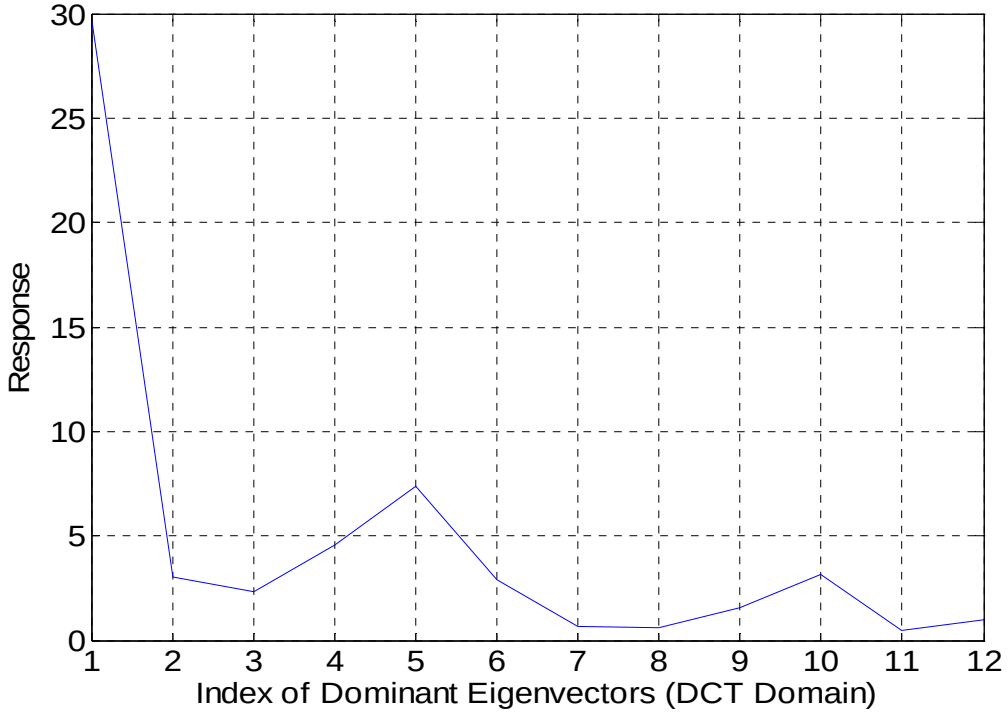

Figure 4.14(b) VIDEO 3: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

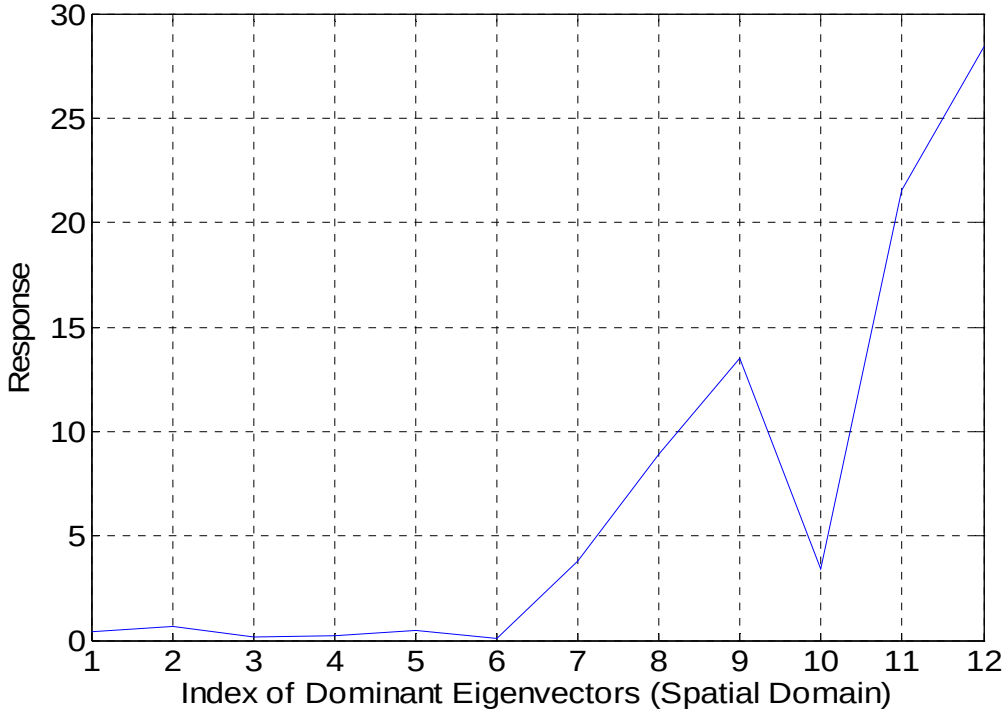

Figure 4.15(a) VIDEO 4: Response of a representative target vector versus the index of the dominant eigenvectors (spatial domain)

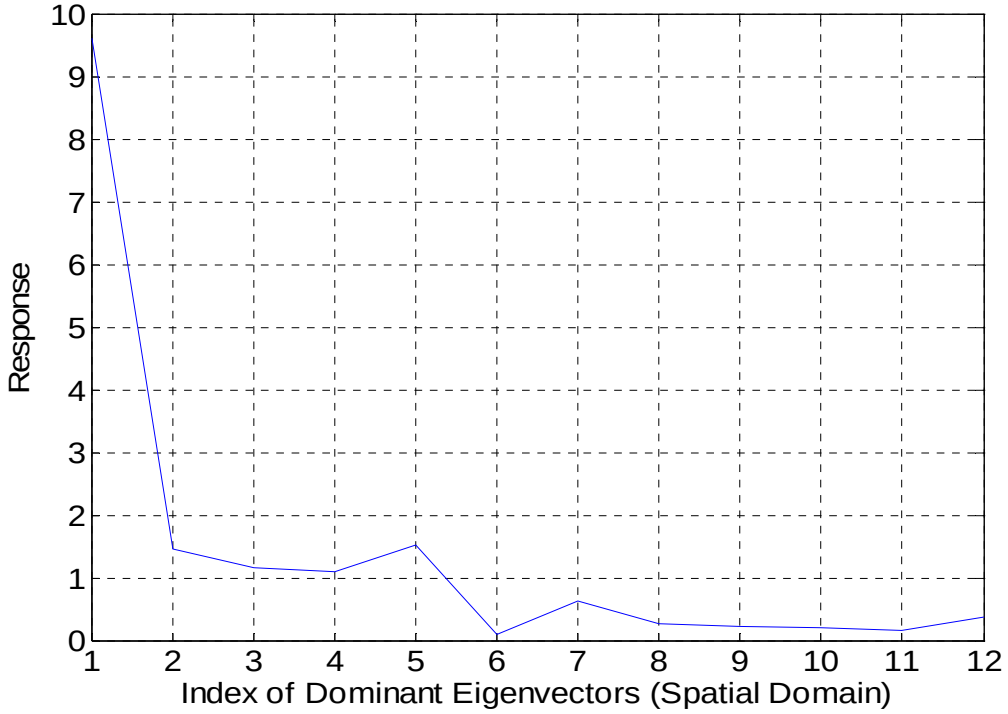

Figure 4.15(b) VIDEO 4: Response of a representative clutter vector versus the index of the dominant eigenvectors (spatial domain)

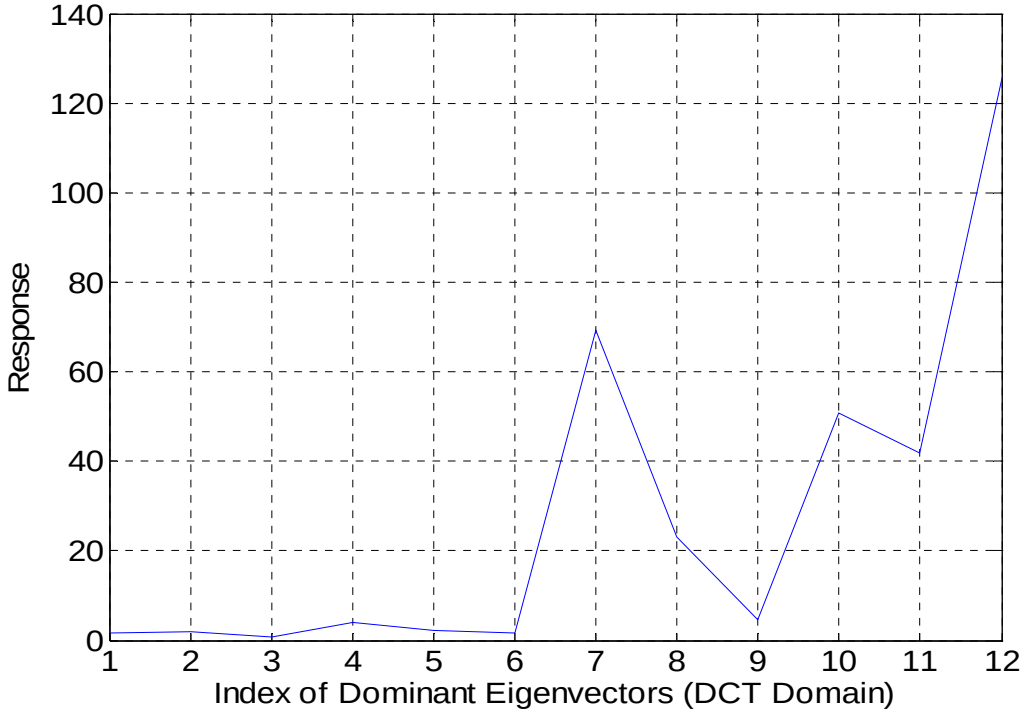

Figure 4.16(a) VIDEO 4: Response of a representative target vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

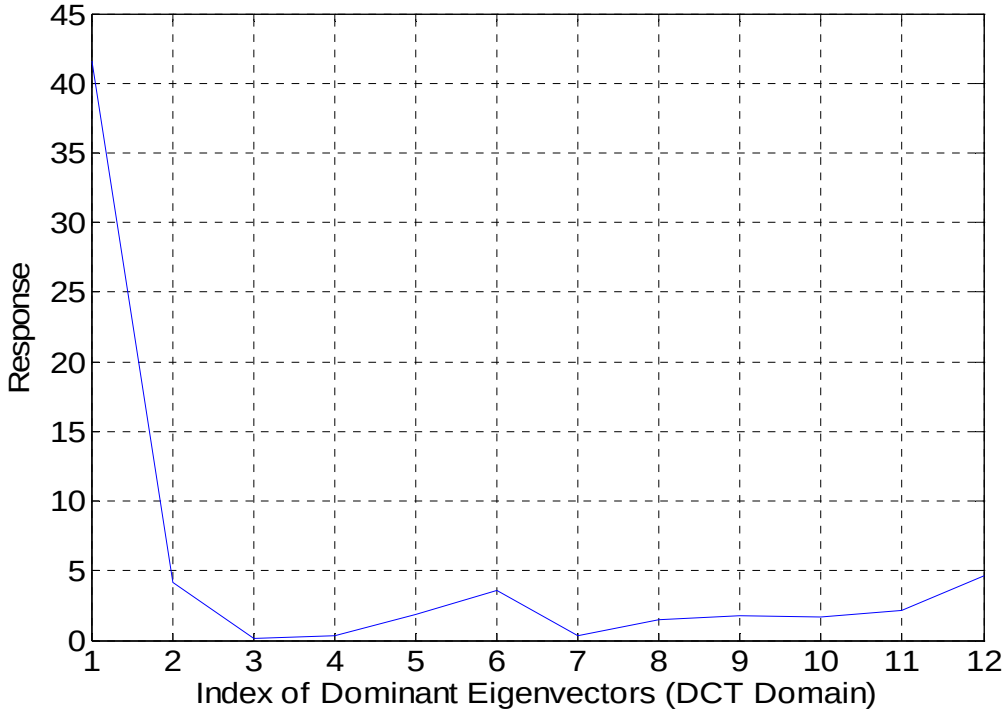

Figure 4.16(b) VIDEO 4: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain

A close look at the plots reveals the following: i) The magnitude of each of the responses, inner products, in the DCT domain is much higher than the corresponding magnitude in the spatial domain, ii) The magnitude of $(p_t - p_c)$ is also much higher in the DCT domain than in the spatial domain. In other words, separation between targets and clutter is also much higher in the DCT domain than in the spatial domain. This means that the requirements on the threshold to decide if a chip is target or clutter can be relaxed considerably. Although the plots shown are for a few randomly chosen data points from the different videos, it was found that the TDRQQCF

consistently produces much larger responses and target-clutter separation than the spatial domain RQQCF for all data points.

Table 4.6 summarizes the recognition accuracy of the spatial domain RQQCF and the TDRQQCF for all the four videos. Each row in the table shows for a particular video, the number of target and clutter chips recognized correctly by the RQQCF and the TDRQQCF.

Table 4.6. Recognition accuracy of the spatial domain RQQCF and the TDRQQCF for all the four videos.

	TARGET		CLUTTER	
	RQQCF	TDRQQCF	RQQCF	TDRQQCF
VIDEO 1	409/409	409/409	409/409	409/409
VIDEO 2	763/763	763/763	763/763	763/763
VIDEO 3	405/405	405/405	405/405	405/405
VIDEO 4	390/391	390/391	390/391	390/391

It is seen that the TDRQQCF retains the excellent recognition accuracy of the spatial domain RQQCF. Also, both the RQQCF and TDRQQCF fail to recognize the same chip in VIDEO 4. This particular chip is shown in Figure 4.17. The reason is that this chip looks more like a clutter

chip than a target chip. For comparison, a sample representative target chip from the video is shown in Figure 4.18.

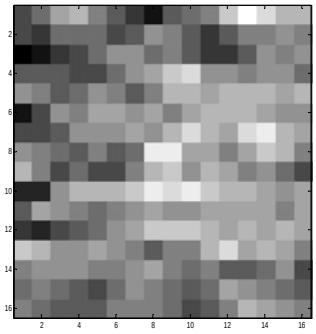

Figure 4.17 Misclassified target chip form VIDEO 4

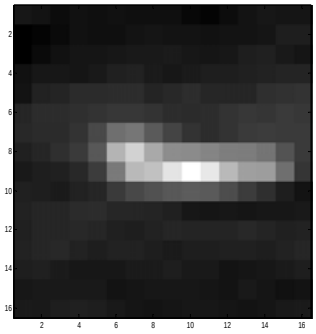

Figure 4.18 Sample representative target chip form VIDEO 4

The overall reduction in computational and storage requirements of the TDRQQCF over the RQQCF is obtained while retaining its recognition accuracy. The RQQCF involves the inversion and Eigenvalue Decomposition (EVD) of large matrices. The computational complexity for each of these operations is of the order $O(n^3)$, where 'n' is the dimensionality of the autocorrelation matrices. On the other hand, by using the TDRQQCF, where compressed representations are used for target and clutter, large savings are obtained. Table 4.7 compares the spatial domain RQQCF with the TDRQQCF in terms of storage and computational complexity.

Table 4.7. Storage and computational complexity of the spatial domain RQQCF versus that for the TDRQQCF, (* from [91]).

	RQQCF	TDRQQCF	% of savings using TDRQQCF*
No. of storage locations for chips	$2 \times M \times \sqrt{n} \times \sqrt{n}$	$2 \times M \times \sqrt{k} \times \sqrt{k}$, $k < n$	75%
No. of storage locations for autocorrelation matrices	$2 \times n \times n$	$2 \times k \times k$	93.75%
Complexity of Inversion**	O(n3)	O(k3)	98%***
Complexity of EVD**	O(n3)	O(k3)	98%***

* For M=409, n=256, k=64; ** # of multiplications; ***approximately

The computational complexity for the 2^j x 2^j DCT is approximately 2j x (2j+1- j - 2), [29 - 31]. From Table 4.7, it can be easily shown that the overall computational complexity including computing the DCT and the storage requirements of the TDRQQCF are still much smaller than the spatial domain RQQCF. In addition, for the TDRQQCF, the storage and computational savings increase as the chip size increases.

In addition to reduced dimensionality, there is another advantage to the TDRQQCF. Often, in practice, in applications of techniques such as the RQQCF, one encounters low rank matrices which give rise to numerical problems and loss in recognition accuracy. This is because the number of data points available for training is very close to or smaller than the dimensionality of each data point leading to poor estimates of class statistics. This effect was observed in Figures 2.7 – 2.10, for the spatial domain RQQCF as the number of training chips was reduced.

On the other hand, TDRQQCF overcomes this problem by reducing the dimensionality of the data points. Figures 4.19 – 4.22 show the plots of recognition accuracy (%) versus the training set size for VIDEOS 1-4.

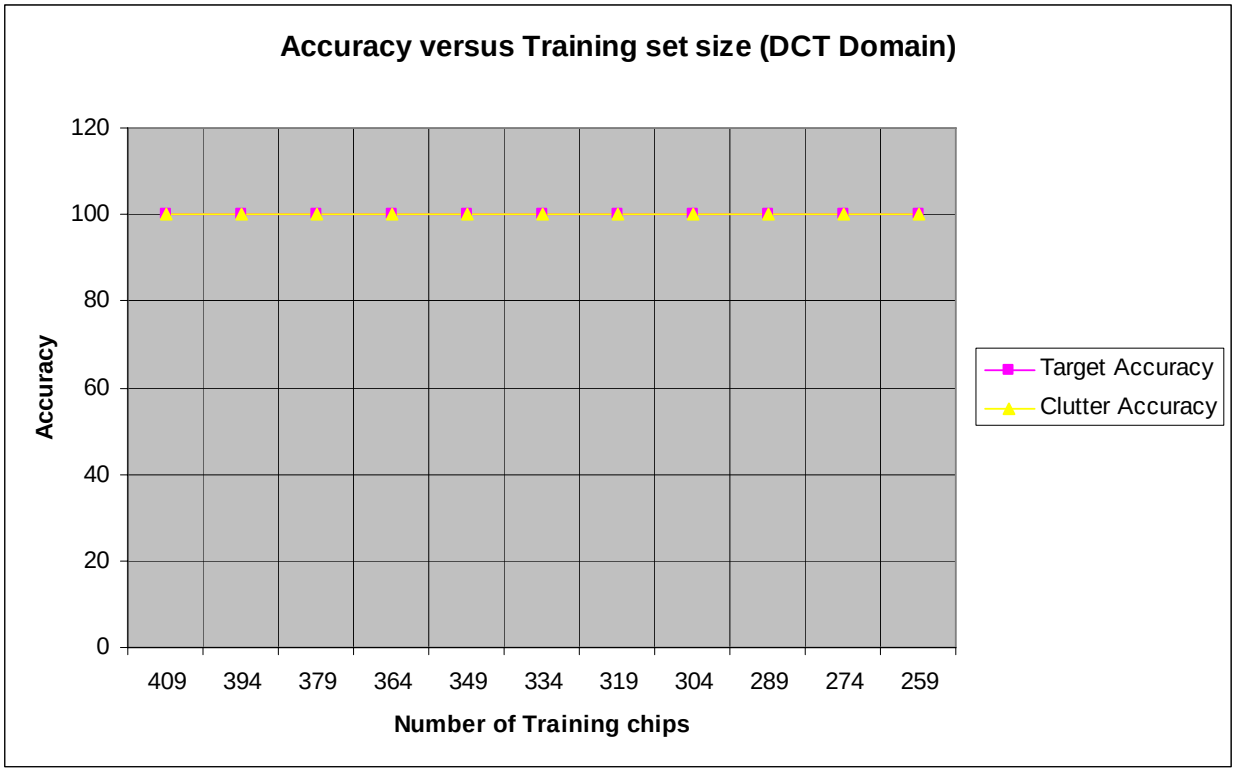

Figure 4.19 VIDEO 1: Accuracy (%) versus training set size

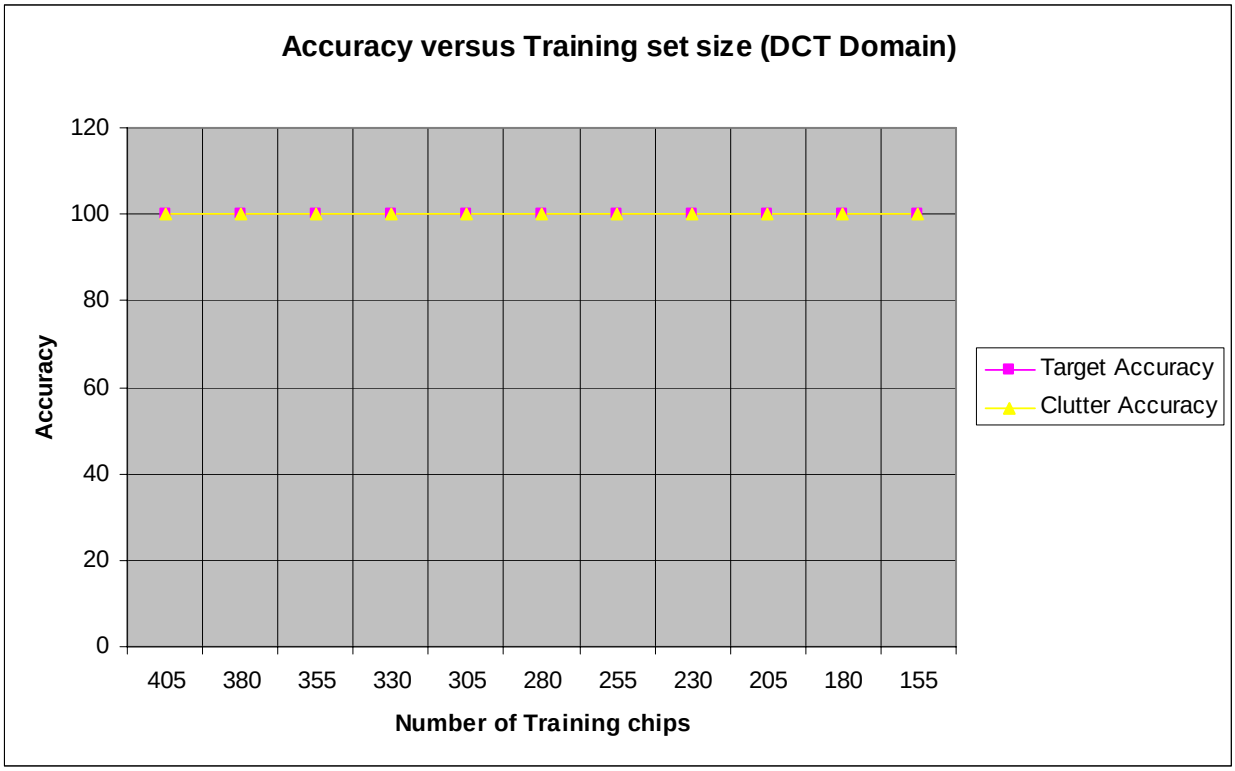

Figure 4.20 VIDEO 2: Accuracy (%) versus training set size

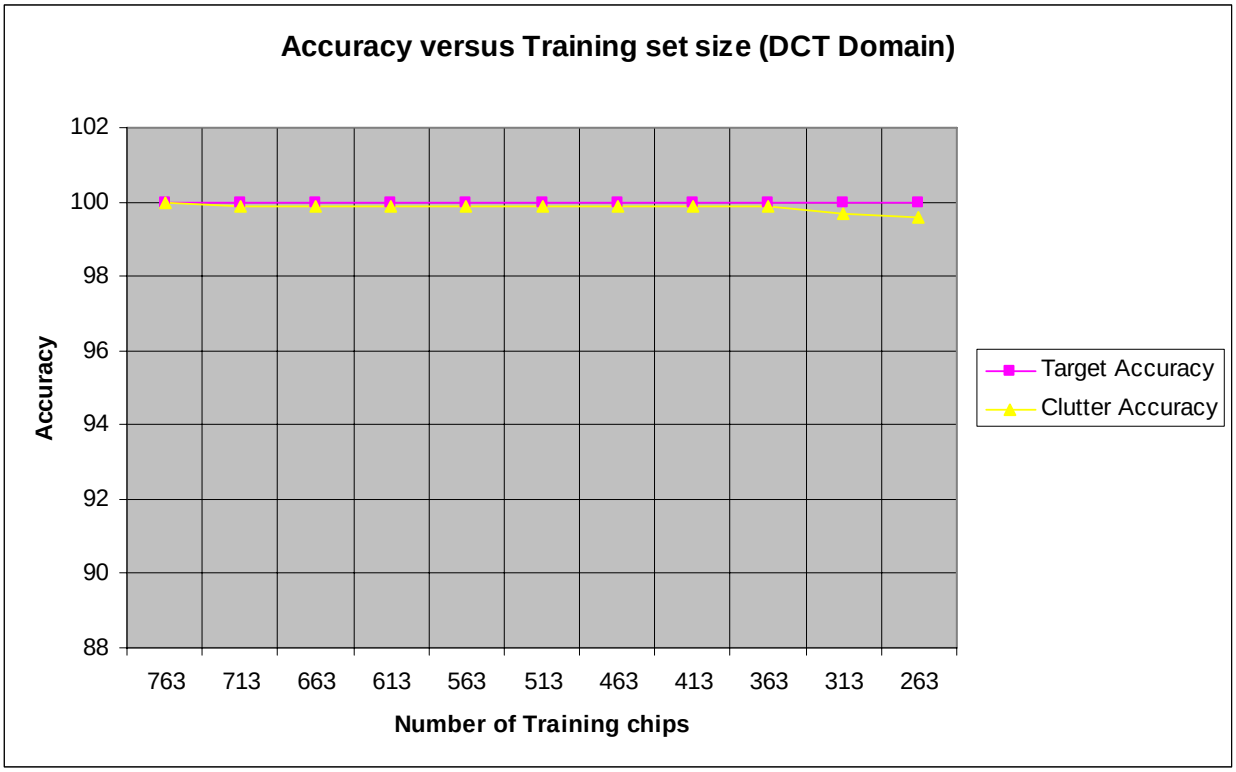

Figure 4.21 VIDEO 3: Accuracy (%) versus training set size

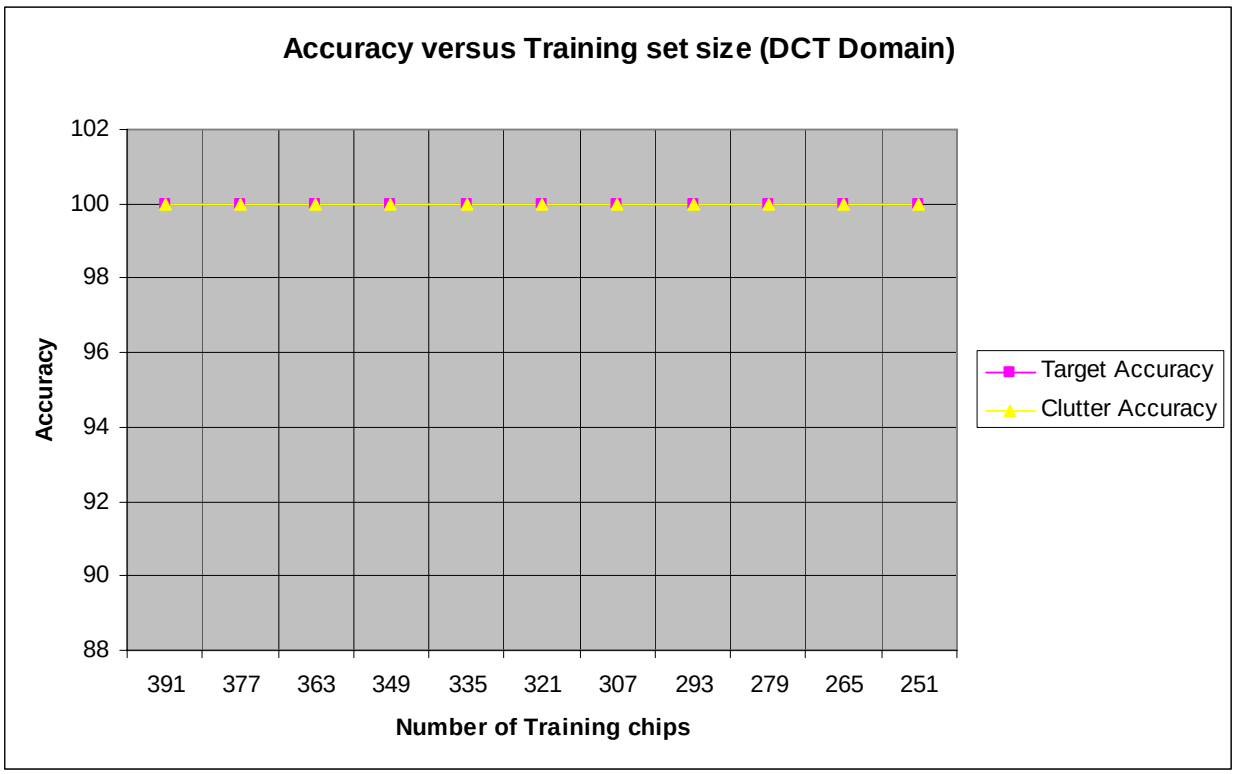

Figure 4.22 VIDEO 4: Accuracy (%) versus training set size

Comparing Figures 4.19 – 4.22 to Figures 2.7 – 2.10 of the spatial domain RQQCF, we can see that the TDRQQCF is able to maintain high recognition accuracy as the number of training chips is reduced.

Comparison of the TDRQQCF with regularization of the RQQCF in the spatial domain

The TDRQQCF presented in the previous section considerably reduced the computational complexity and storage requirements, by compressing the target and clutter data used in designing the QCF. In addition, the TDRQQCF approach was able to produce larger responses when the filter was correlated with target and clutter images. This was achieved while maintaining the excellent recognition accuracy of the original spatial domain RQQCF algorithm. The computation of the RQQCF and the TDRQQCF involve the inverse of the term $A_1 = (R_x+R_y)$ where R_x and R_y are the sample autocorrelation matrices for targets and clutter respectively. It can be conjectured that the TDRQQCF approach is equivalent to regularizing A_1. A common regularization approach involves performing the Eigenvalue Decomposition (EVD) of A_1, setting some small eigenvalues to zero, and then reconstructing A_1, which is now better conditioned. In this section, this regularization approach is investigated, and compared to the TDRQQCF. Sample simulation results show that these approaches do not produce the same results; in fact, the regularization actually degrades the RQQCF performance while the TDRQQCF maintains it.

Simulation Results

To regularize $(R_x + R_y)^{-1}$, we perform the EVD of $(R_x + R_y)^{-1}$ and set a small number of eigenvalues (of small magnitudes) to zero, and then reconstruct. The idea is that this "noise removal" procedure will by itself result in larger responses and target-clutter separation.

As the sample results will show, the two techniques are different and produce different results. The TDRQQCF produces larger responses and target-clutter separation while maintaining the accuracy of the spatial domain RQQCF. In addition, it results in considerable savings in storage and computation while also overcoming the problems of small training sets that are often encountered in practice. On the other hand, the regularization technique results in smaller responses and target-clutter separation and in some cases reduced accuracy.

Figure 4.23 shows a sample frame from the video used for simulations.

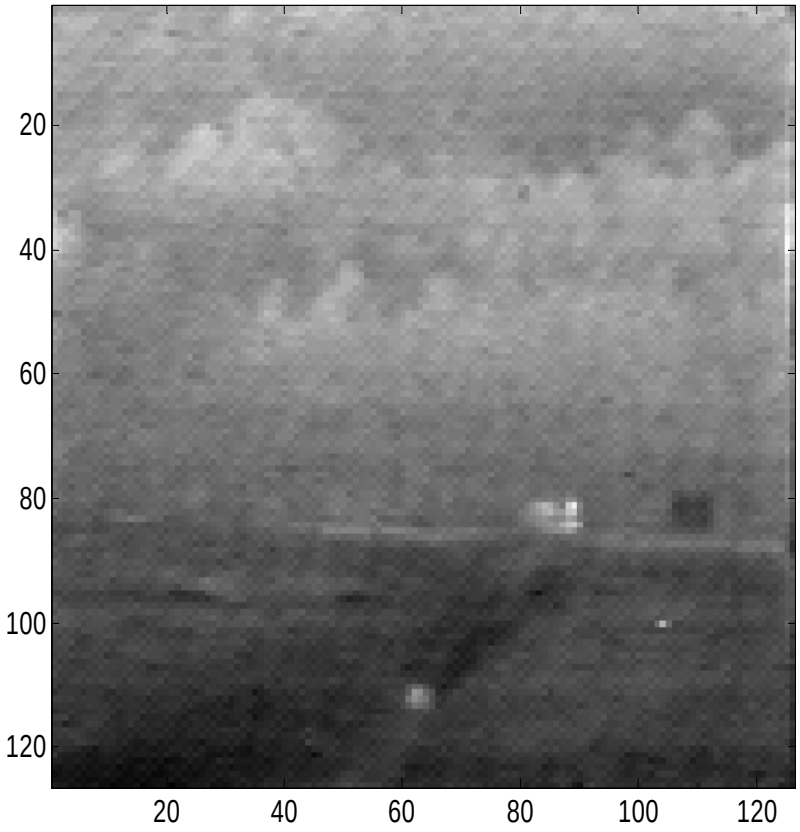

Figure 4.23 Sample frame from VIDEO 1

Results of original RQQCF formulation in the spatial domain

lambda_orig (dominant eigenvalues) = -0.9975 -0.9641 -0.9559 -0.9378 -0.9283 -0.9221 0.9934 0.9938 0.9962 0.9971 0.9981 0.9985

Figures 4.24(a) and 4.24(b) show the response of target data point and clutter data point respectively. As shown in Chapter 2, all target and clutter points were identified correctly.

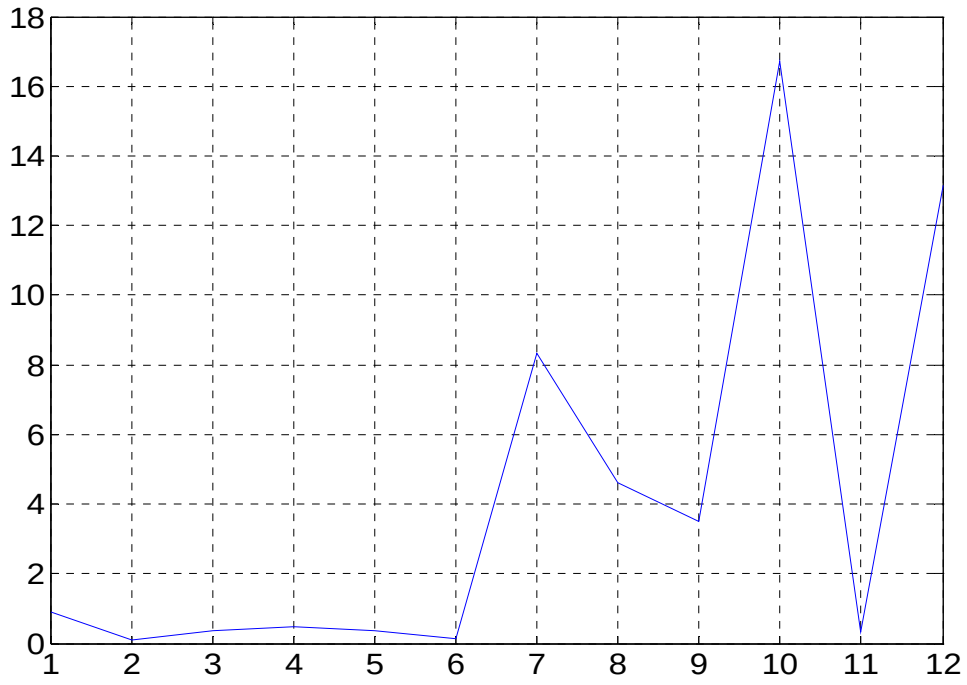

Figure 4.24(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors (spatial domain)

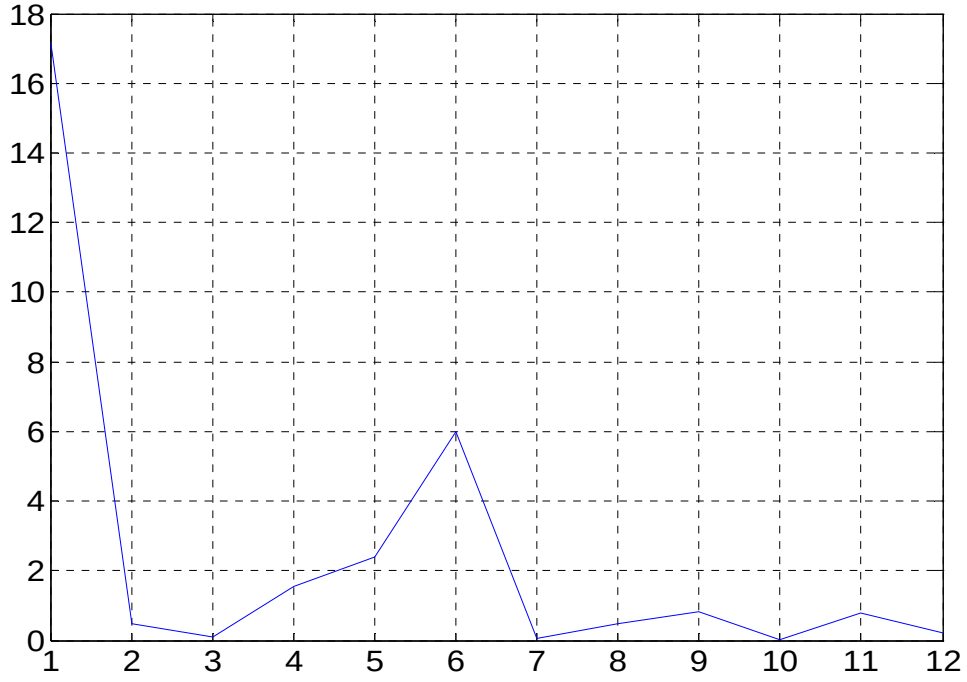

Figure 4.24(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors (spatial domain). (The condition number of A, cond(A)=1.3027e+006)

Results of TDRQQCF for the same data points used in the RQQCF

lambda_orig (dominant eigenvalues) = -0.9930 -0.8648 -0.7825 -0.7642 -0.7472 -0.6427

0.9642 0.9722 0.9746 0.9878 0.9943 0.9952

Figures 4.25(a) and 4.25(b) show the response of target data point and clutter data point respectively. As shown in Chapter 4, all target and clutter points were identified correctly.

Figure 4.25(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain.

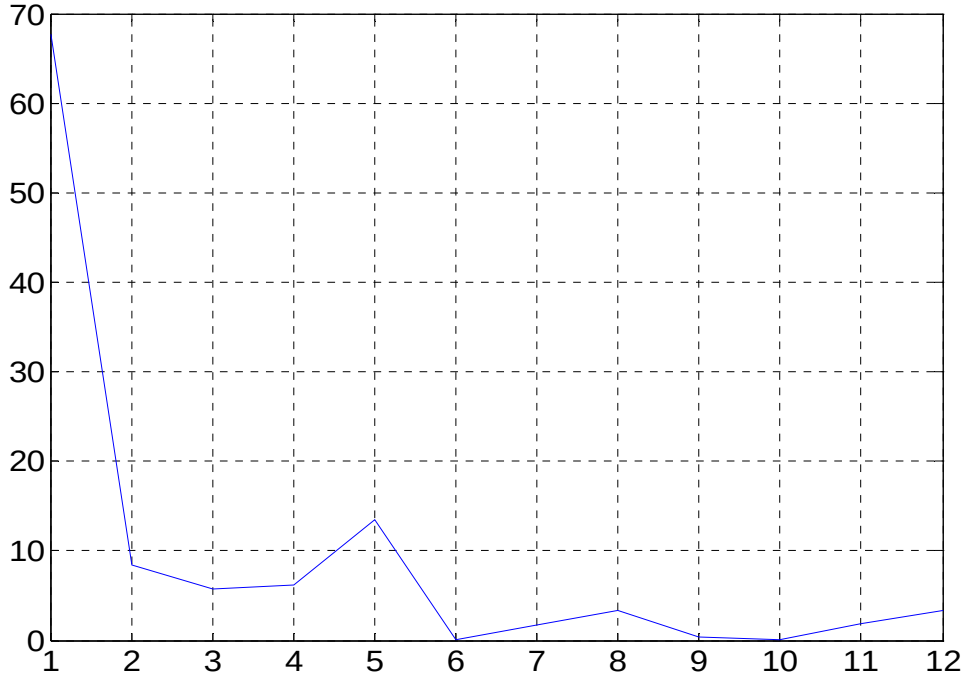

Figure 4.25(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the truncated chips (8x8) in the DCT domain. (The condition number of A, cond(A) = 6.1722e+004)

Comparing the responses in Figures 4.24(a) and 4.25(a), it can be seen that the magnitude of the responses in the DCT domain is higher than the spatial domain RQQCF. Similar results are observed for clutter points (Figures 4.24(b) and 4.25(b)). The condition number of A has decreased.

We now compare the previous TDRQQCF result where the transformed 16x16 chips are truncated to 8x8 to the case where the transformed chips size is kept the same (16x16) but the DCT coefficients outside the top-left 8x8 part of the coefficient matrix are set to zero.

lambda_orig (dominant eigenvalues) = -0.9930 -0.8648 -0.7825 -0.7642 -0.7472 -0.6427 0.9642 0.9722 0.9746 0.9878 0.9943 0.9952

Figures 4.26(a) and 4.26(b) show the response of target data point and clutter data point respectively. All target and clutter points were identified correctly. These results exactly match those in figures 4.25(a) and 4.25(b). The condition number of A is very high (the matrix is close to singular).

Figure 4.26(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors derived from the chips (8x8) in the DCT domain with coefficients set to zero instead of truncation.

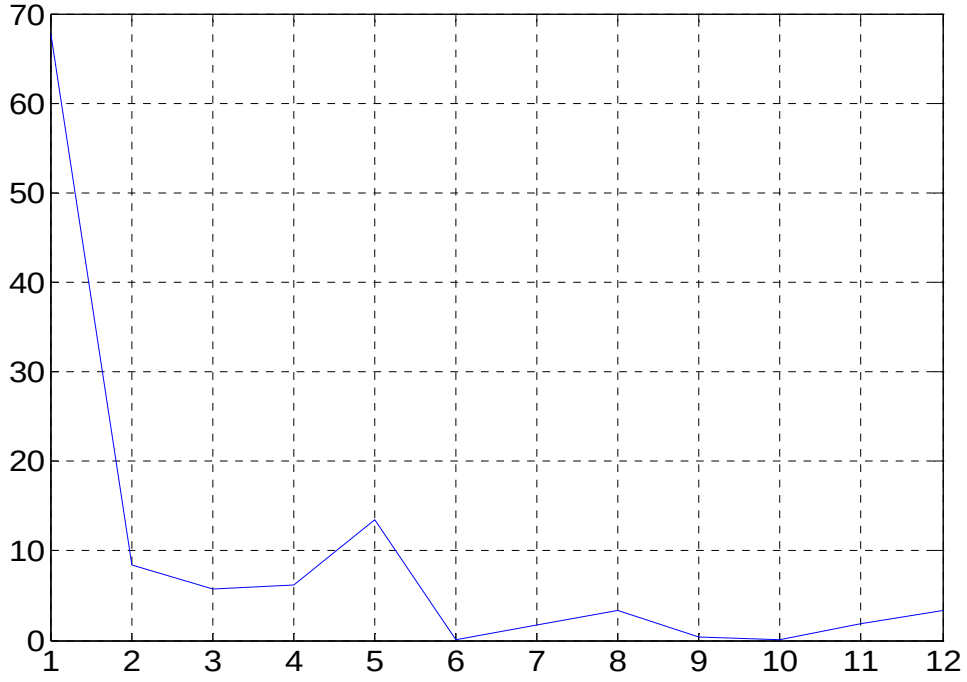

Figure 4.26(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors derived from the chips (8x8) in the DCT domain with coefficients set to zero instead of truncation. (The condition number of A, cond(A) = Inf)

Results of original RQQCF formulation in the spatial domain with the regularization of $A_1 = (R_x + R_y)^{-1}$. The EVD of A_1 is performed, the smallest 5 eigenvalues are set to zero, and A_1 is reconstructed. This 'regularized' A_1 is used to form the matrix A as in equation 8 of Chapter 2.

lambda_orig (dominant eigenvalues) = -0.9472 -0.9269 -0.9225 -0.9181 -0.9127 -0.9025 0.9630 0.9691 0.9701 0.9792 0.9805 0.9850

Figures 4.27(a) and 4.27(b) show the response of a target data point and a clutter data point respectively. Two of the target and clutter points were identified incorrectly, i.e. they were misclassified.

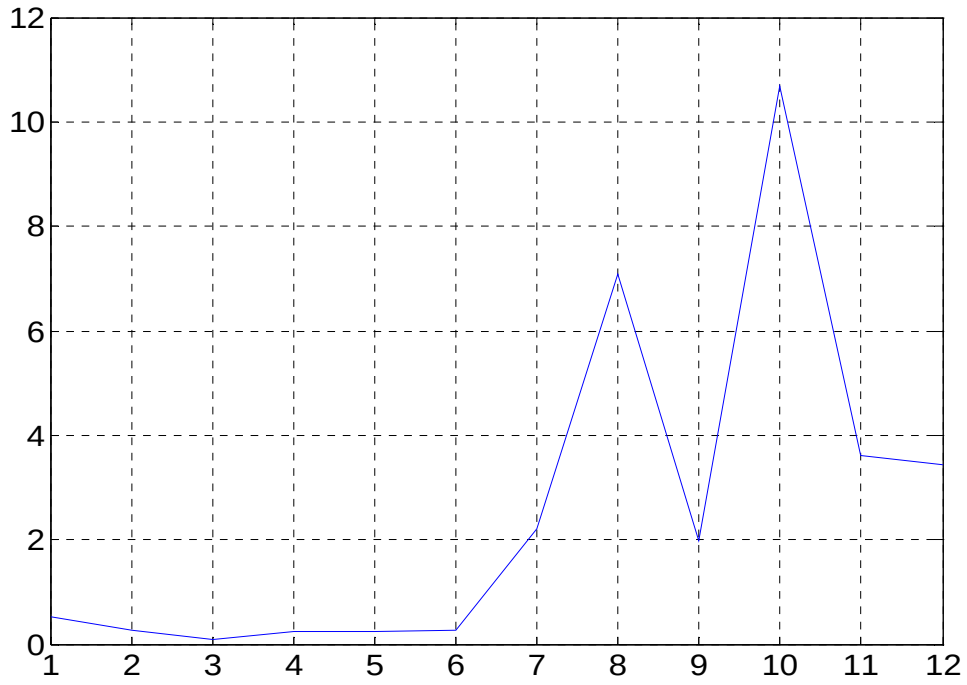

Figure 4.27(a) VIDEO 1: Response of a representative target vector versus the index of the dominant eigenvectors of the regularized spatial domain RQQCF approach.

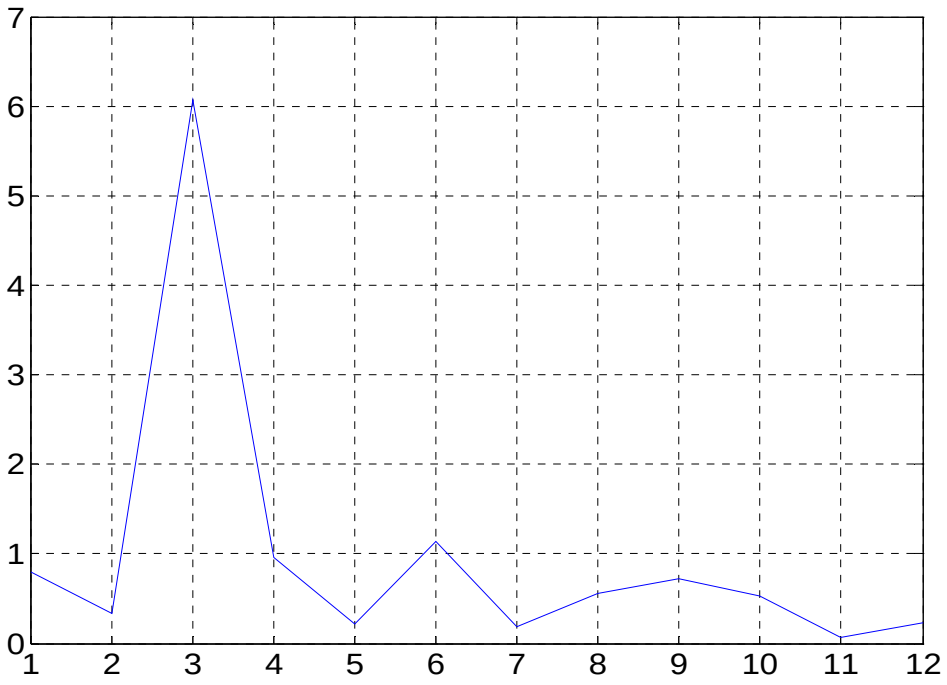

Figure 4.27(b) VIDEO 1: Response of a representative clutter vector versus the index of the dominant eigenvectors of the regularized spatial domain RQQCF approach. (The condition number of A, cond(A) = 1.3027e+006)

In addition to loss of accuracy, the magnitudes of the responses have actually decreased compared to the original RQQCF approach. The condition number of A has also increased as expected.

It is seen that as more and more eigenvalues are set to zero, the performance of the regularized RQQCF worsens. Not only do the magnitude of the responses drop compared to the spatial domain RQQCF, but the accuracy of the QCF also drops.

Summary

We can conclude that the TDRQQCF and the regularized RQQCF are not the same. They produce different results. The regularization procedure, results in smaller responses and target-clutter separation and in some cases reduced accuracy. The TDRQQCF produces larger responses and target-clutter separation while maintaining the accuracy of the spatial domain RQQCF. In addition, it results in considerable savings in storage and computation while also overcoming the problems of small training sets that are often encountered in practice.

TDRQQCF Summary

A novel Transform Domain Rayleigh Quotient Quadratic Correlation Filter (TDRQQCF) is proposed. The improved performance of the TDRQQCF results from compressing the data in the transform domain. Consequently, this leads to considerable reduction in computational complexity and storage requirements over the spatial domain RQQCF technique while retaining excellent recognition accuracy. It is worthwhile to note another advantage of the new technique, namely, the TDRQQCF acts as a low pass filter that removes noise. Consequently, the separability between targets and clutter improves. In addition, the method overcomes the problems of small training sets that are often encountered in practice. One can also argue that changes to the QCF coefficients due to new data can now be obtained by direct inversion and EVD

methods of much reduced computational complexity using the TDRQQCF. This is confirmed by extensive simulation results. Sample results are given.

CHAPTER FIVE: A TWO DIMENSIONAL RQQCF

In the original RQQCF formulation, the target and clutter chips which are two-dimensional (2D) are first converted to a one-dimensional (1D) vector before synthesizing the filter. The TDRQQCF as described in the previous section combined the detection/recognition problem with compression. In addition, it also alleviated the problem of a relatively small training set. The "tolerance" to a small training set is governed by how much compression can be achieved without compromising the accuracy of the QCF. In some situations, the training set can be so small relative to the dimensions of the data points that even the TDRQQCF may not be an effective solution. Consider for example, a face recognition problem using the ORL/ATT database. This database contains 10 facial images each of 40 individuals. The size of each image is 112x92. It is very easy to see that in this case, there are very few observations – 10, to estimate the class autocorrelation matrices (of size 10304x10304) – even with the TDRQQCF approach. In the 2D formulation, the aim is to keep these chips as 2D. It is seen that this approach is able to successfully address the problem of small training sets. Also, the dimensions of the filter, and therefore the storage and computational requirements are much reduced.

The Trace formulation of the 2DRQQCF

The QCF coefficient matrix T is assumed to take the form,

$$T = \sum_{i=1}^{n} \underline{w_i}\,\underline{w_i^T} \tag{33}$$

where, $\underline{w_i}, 1 \le i \le n$, form an orthonormal basis set. The objective of the technique is to determine these basis functions such that the separation between the two classes, say X and Y, is maximized. The output of the QCF to an input image U is given by

$$\varphi = trace(U^T T U) \tag{34}$$

The objective is to maximize the ratio,

$$J(\underline{w}) = \frac{E_1\{\varphi\} - E_2\{\varphi\}}{E_1\{\varphi\} + E_2\{\varphi\}} = \frac{\sum_{i=1}^{n} \underline{w_i}(R_x - R_y)\underline{w_i^T}}{\sum_{i=1}^{n} \underline{w_i}(R_x + R_y)\underline{w_i^T}} \tag{35}$$

where, $E_j\{.\}$ is the expectation operator over the jth class, and R_x and R_y are the correlation matrices for targets and clutter respectively, which are calculated as follows,

$$R_x = E_X(UU^T) \tag{36}$$

$$R_y = E_Y(UU^T) \tag{37}$$

Taking the derivative of Equation 35 with respect to $\underline{w_i}$, we get

$$(R_x + R_y)^{-1}(R_x - R_y)\underline{w_i} = \lambda_i \underline{w_i} \tag{38}$$

Let,

$$A = (R_x + R_y)^{-1}(R_x - R_y) \tag{39}$$

Thus $\underline{w_i}$ is an eigenvector of A with eigenvalue λ_i. It should be noted that $J(\underline{w})$ is in the form of a Rayleigh Quotient which is maximized by the dominant eigenvector of A.

It can be shown that the trace formulation of the 2DRQQCF as shown above amounts to treating each column of a target chip as a target by itself and each column of a clutter chip as clutter. Therefore, the response of a chip to the QCF designed using the above formulation is the sum of the responses of its columns.

If the output of the 2DRQQCF is defined as $\varphi = trace(UTU^T)$, the trace formulation amounts to treating each row of a target chip as a target by itself and each row of a clutter chip as clutter. Therefore, the response of a chip to the QCF designed using the above formulation is the sum of the responses of its rows.

Simulation Results

The 2DRQQCF was used in the context of a facial recognition problem. To recognize/classify two images or groups of images, we can call one image or group of image as "targets" while the the other image or images can be called "clutter". Thus, we can essentially treat the facial

recognition or classification problem as an ATR problem and use the 2DRQQCF as described in the previous section.

To demonstrate the performance of the 2DRQQCF, simulation results are presented on the ORL/ATT Face Recognition Database. This database contains 10 images each of 40 individuals. The 10 images per person are representative of different illumination conditions, different facial expressions and facial details. A sample set of faces from the database is shown in Figure 5.1. Each image is of size 112x92.

Figure 5.1 Sample images from the ORL/ATT Facial Recognition Database

Various scenarios were considered where different numbers of images were chosen in the training phase to synthesize the 2DRQQCF and the corresponding remainders of images were chosen for testing. In addition, the recognition/classification was done between 1) two individuals and 2) two groups of individuals.

The training phase for classification of two individuals, say X and Y proceeds as follows:

1. A certain number of images, say N, are chosen from the M available images each of X and Y.
2. The autocorrelation matrices, R_x and R_y are calculated as shown in Equation 36 and Equation 37.
3. The matrix A given in Equation 39 is computed.
4. The EVD of A is performed to obtain the QCF coefficients.

In the testing phase,

1. The M-N test images of each individual are correlated with the QCF coefficients.
2. The net response of each image to the QCF is calculated. If the net response is positive, the test image is recognized or classified as belonging to the target class, say X, else, it is classified as belonging to the clutter class, say Y.

It is easy to see that a straightforward extension of the procedure outlined above can be used to recognize or classify a group of individuals.

Case 1 Distinguishing between two individuals - S1 (target) and S2 (clutter)

(a) All the images for each individual are used for training. These same images are also used for testing, i.e. M=N=10. If there are false matches in this case, then we can be sure that the accuracy will only worsen if we use different sets of images for training and testing. Figure 5.2 shows the eigenvalue distribution of A for the above scenario.

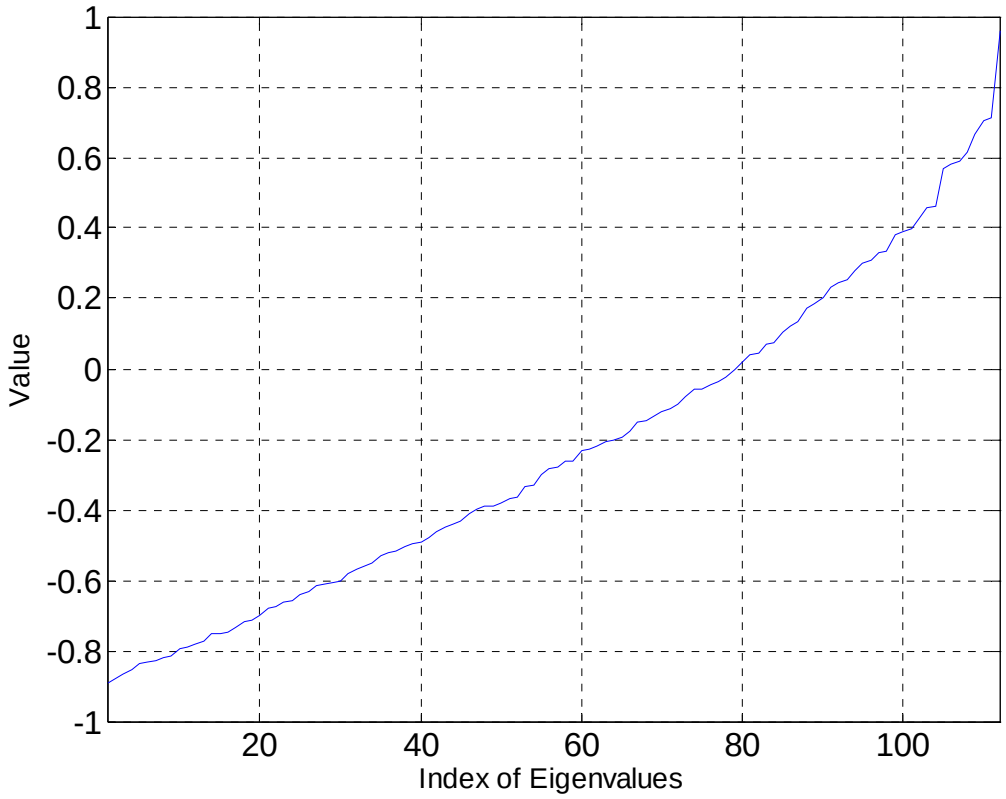

Figure 5.2 Distribution of eigenvalues for M=N=10

Figure 5.3 and 5.4 show the response of the 10^{th} image from S1 and the 10^{th} image from S2 to the 2DRQCCF. It is clear to see that the image corresponding to Figure 5.3 belongs to S1 while the image corresponding to Figure 5.4 belongs to S2.

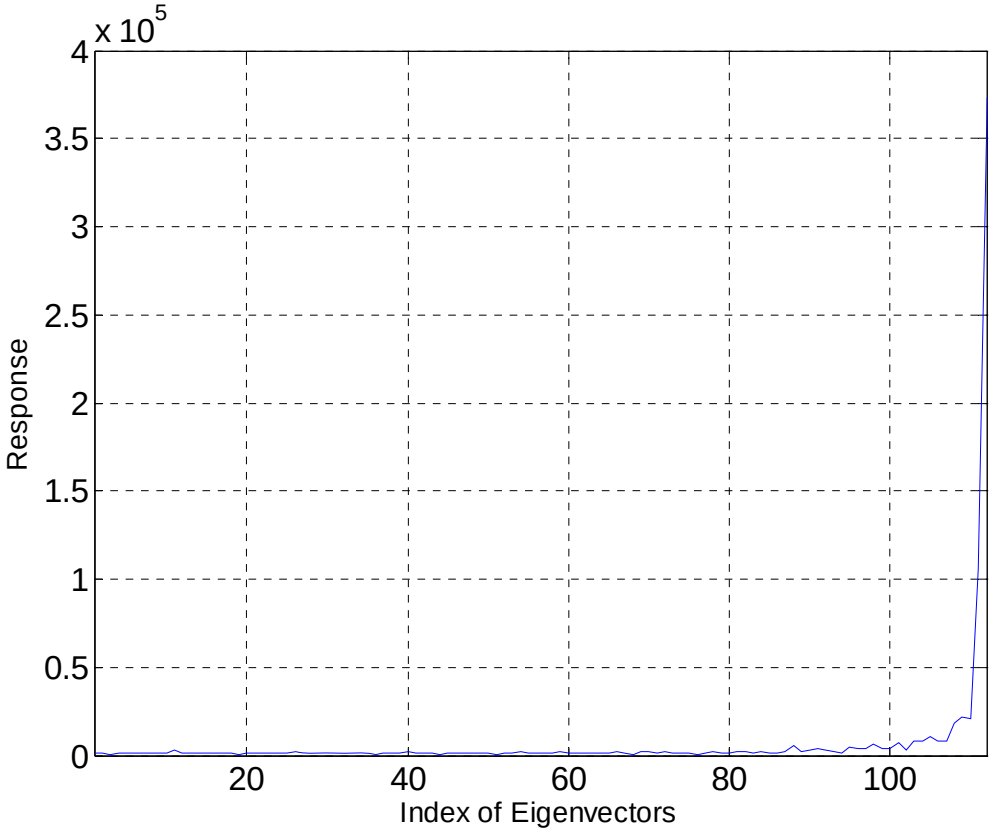

Figure 5.3 Response of point no. 10 from S1 to the 2DRQQCF

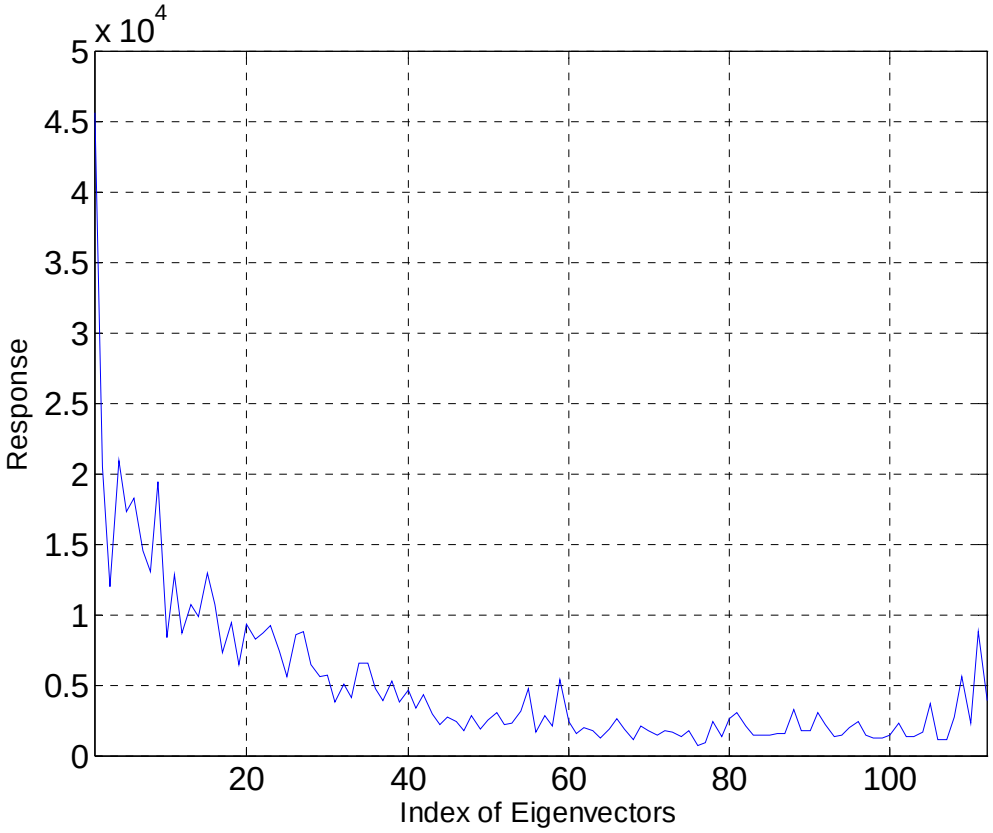

Figure 5.4 Response of point no. 10 from S2 to the 2DRQQCF

(b) Five images for each individual are used for training. The remaining five images of each are used for testing, i.e. M=N=5. Figure 5.5 and 5.6 show the response of the 5th image from S1 and the 5th image from S2 to the 2DRQCCF. It is clear to see that the image corresponding to Figure 5.5 belongs to S1 while the image corresponding to Figure 5.6 belongs to S2.

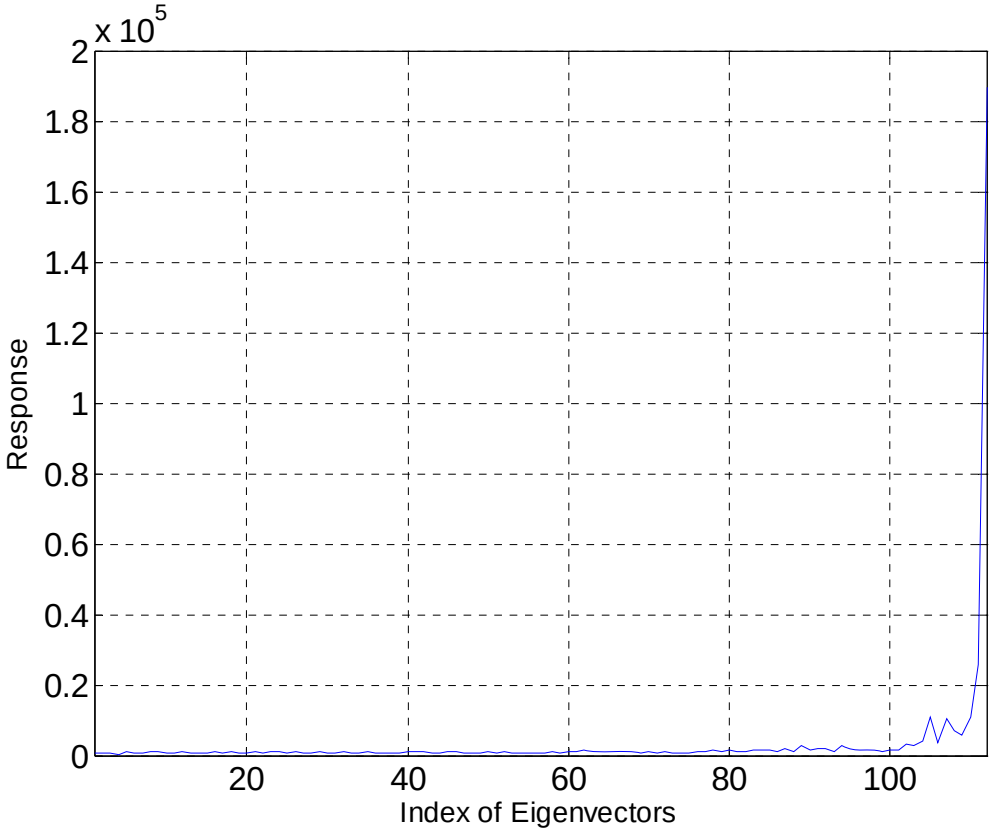

Figure 5.5 Response of point no. 5 from S1 to the 2DRQQCF

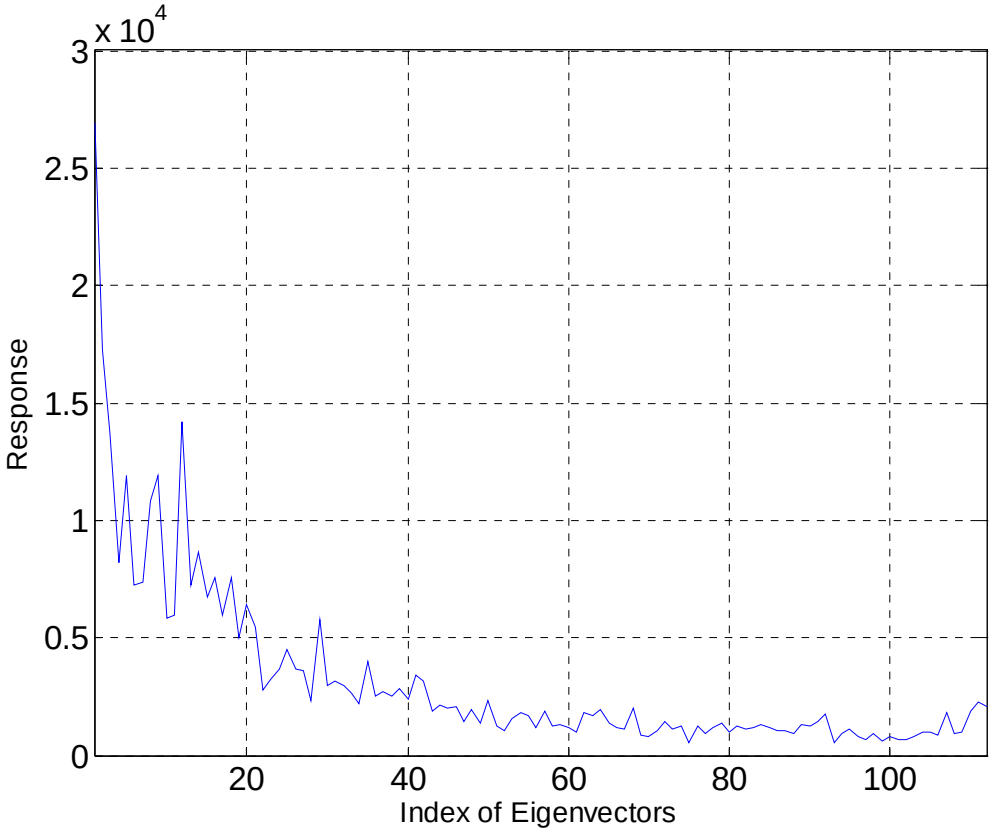

Figure 5.6 Response of point no. 5 from S2 to the 2DRQQCF

Case 2 Distinguishing between two sets of individuals

T = [S1; S2; S3; S4] (target) and C = [S6; S7; S8; S9] (clutter). Nine images for each individual are used for training. The remaining eight images (one of each) are used for testing, i.e. M=40, N=36.

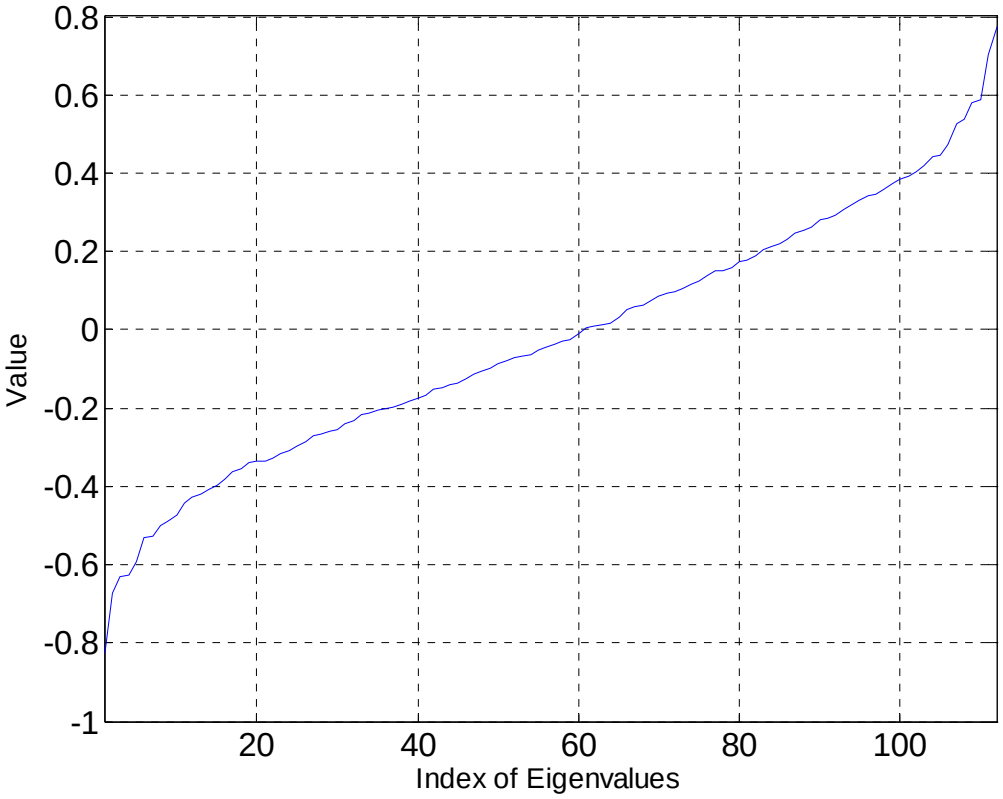

Figure 5.7 Distribution of eigenvalues for M=40, N=36

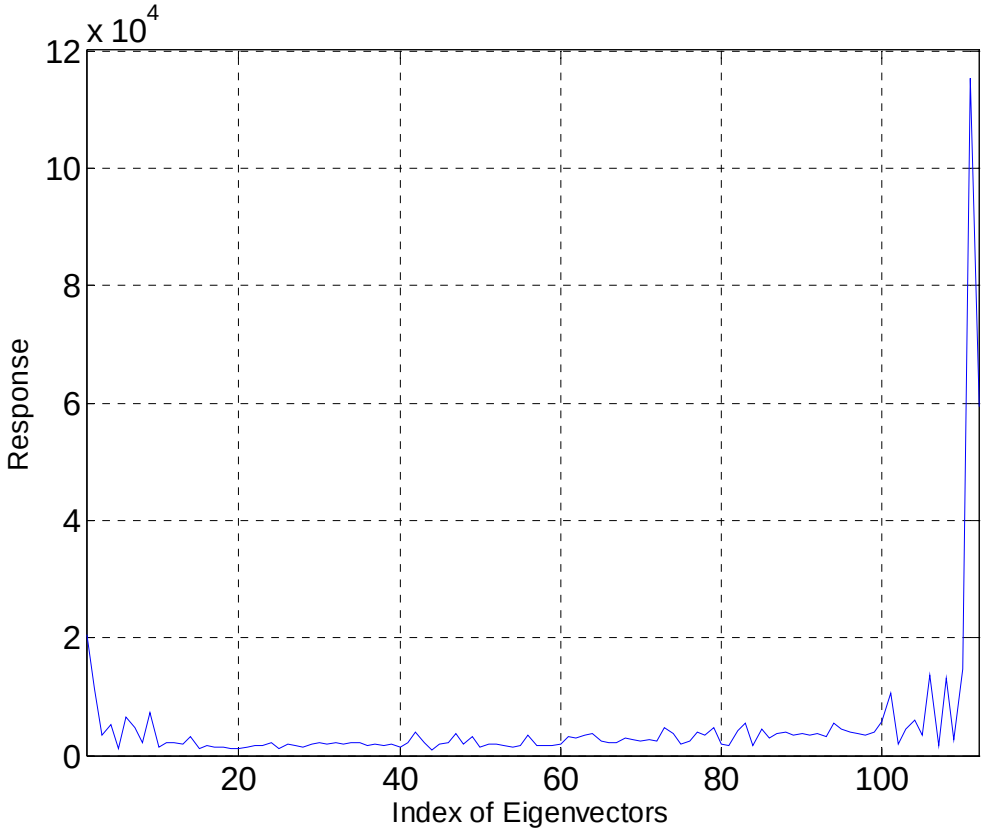

Figure 5.8 Response of point no. 37 from T to the 2DRQQCF

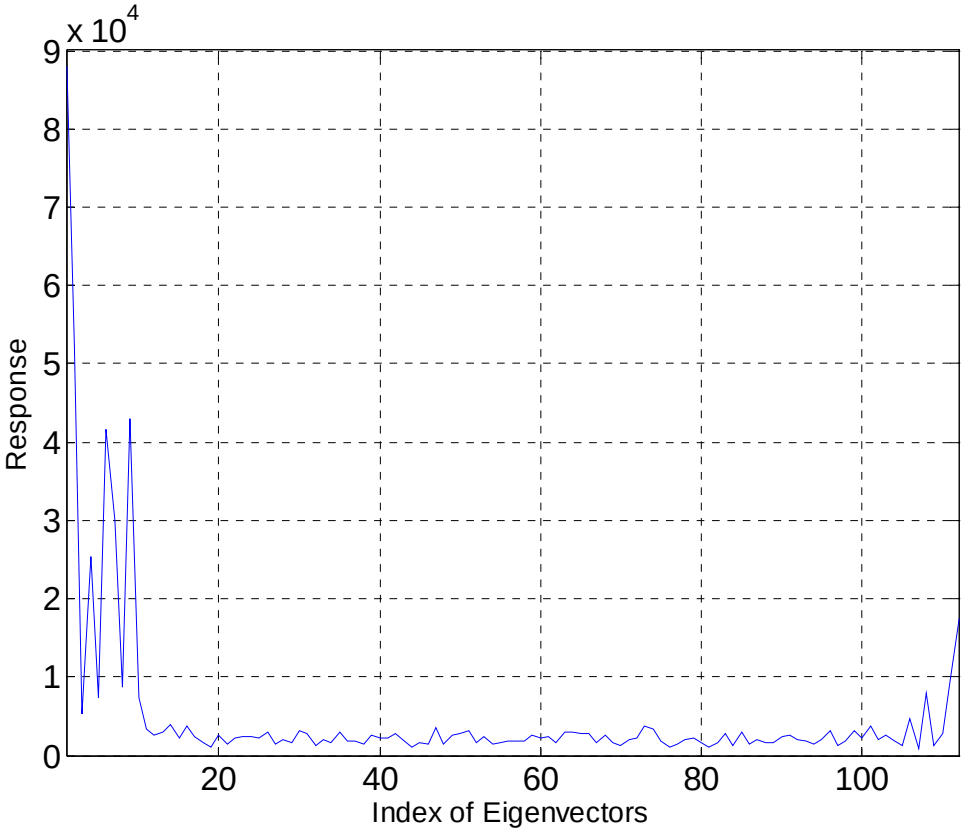

Figure 5.9 Response of point no. 37 from C to the 2DRQQCF

It was seen that in all the cases listed above, all testing images were successfully recognized.

Summary

A novel 2D formulation of the RQQCF was presented. It was shown that this approach further saves on storage requirements and computational complexity when compared to the RQQCF and

TDRQQCF. Simulation results using a facial recognition database confirmed that additionally, the 2DRQQCF has excellent recognition accuracy.

CHAPTER SIX: CONCLUSION

Efficient techniques based on the Rayleigh Quotient Quadratic Correlation Filter (RQQCF) technique were presented. The first technique, called the Optimal Adaptive Eigenvalue Decomposition Technique (OAEVD) was an efficient adaptive technique, in terms of speed and computational complexity without sacrificing the accuracy that utilizes the old EVD to search for the new EVD. It avoids matrix inversion and direct EVD, thus providing substantial computational savings. In addition, the OAEVD adaptively updates any particular set of eigenvalues and corresponding eigenvectors of interest. In our application, these are the dominant eigenpairs.

Secondly, a Transform Domain Rayleigh Quotient Quadratic Correlation Filter (TDRQQCF) was proposed. Using simulation results on Infrared (IR) data, it was shown that the TDRQQCF reduces the storage requirements and computational complexity significantly. Additionally, for situations where the dimension of the data points is large compared to the number of data points available to estimate the autocorrelation matrices of target and clutter, the TDRQQCF provides a way to alleviate the problem of rank deficient matrices leading to numerical problems. This is achieved by compressing the data and thereby reducing the size of the data points.

Finally, a Two Dimensional RQQCF (2DRQQCF) was presented. By the treating the target and clutter chips as 2D objects, as opposed to converting them to vectors by lexicographical ordering

of columns, the 2DRQQCF has the potential to further save on computations and storage. Using a facial recognition database – the ORL Database, it is shown that the approach also has excellent recognition accuracy. It is also extremely useful in cases where the training sets are very small, where even the TDRQQCF approach has limitations because of accuracy concerns as data is compressed beyond acceptable limits.

Future work includes testing the 2DRQQCF over a large number of images from other databases and a more extensive study of its performance. In addition, future work also includes applying the techniques described in the previous chapters to other applications such as fingerprint recognition.

LIST OF REFERENCES

1. A. K. Suykens, G. Horvath, S. Basu, C. Micchelli, J. Vandewalle (Eds.) Advances in Learning Theory: Methods, Models and Applications, NATO Science Series III: Computer & Systems Sciences, Volume 190, IOS Press Amsterdam, 2003.
2. M. I. Schlesinger, V. Hlavác, Ten Lectures on Statistical and Structural Pattern Recognition, Kluwer Academic Publishers, 2002.
3. D. J. Hand, H. Mannila and P. Smyth, Principles of Data Mining, MIT Press, August 2001.
4. A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley & Sons, 2001.
5. T. Hastie, R. Tibshirani, and J. Fridman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, 2001.
6. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification (2nd ed.), John Wiley and Sons, 2001.
7. S. Raudys, Statistical and Neural Classifiers, Springer, 2001.
8. G.J. McLachlan and D. Peel, Finite Mixture Models, New York: Wiley, 2000.
9. M. Friedman and A. Kandel, Introduction to Pattern Recognition, statistical, structural, neural and fuzzy logic approaches, World Scientific, Signapore, 1999.
10. D. J. Hand, J. N. Kok and M. R. Berthold, Advances in Intelligent Data Analysis, Springer Verlag, Berlin, 1999.
11. B. Schölkopf, C. J. C. Burges, and A. J. Smola, Advances in Kernel Methods, Support Vector Learning MIT Press, Cambridge, 1999.
12. S. Theodoridis, K. Koutroumbas, Pattern recognition, Academic Press, 1999.

13. A. Webb, Statistical pattern recognition, Oxford University Press Inc., New York, 1999.
14. M. Berthold, D. J. Hand, Intelligent Data Analysis, An Introduction, Springer-Verlag, 1999.
15. V. Cherkassky and F. Mulier, Learning from data, concepts, theory and methods, John Wiley & Sons, New York, 1998.
16. L. Devroye, L. Gyorfi, G.Lugosi, A Probabilistic Theory of Pattern Recognition, Springer-Verlag New York, Inc.1996.
17. E. Gose, R. Johnsonbaugh, S. Jost, Pattern recognition and image analysis, Pretice Hall Inc., 1996.
18. J. Schurmann, Pattern classification, a unified view of statistical and neural approaches, John Wiley & Sons, New York, 1996.
19. V.N. Vapnik, The Nature of Statistical Learning Theory, Springer,1996.
20. B. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.
21. C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
22. D. Paulus and J. Hornegger, Pattern Recognition and Image Processing in C++, Vieweg, Braunschweig, 1995.
23. R. Schalkhoff, Pattern Recognition, statistical, structural and neural approaches, John Wiley and Sons, New York, 1992.
24. G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley and Sons, New York, 1992.
25. B. V. Dasarathy, Nearest neighbor(nn) norms: NN pattern classification techniques, IEEE Computer Society Press, Los Alamitos, 1991.

26. S.M. Weiss and C.A. Kulikowski, Computer Systems that Learn, Morgan Kaufmann, San Mateo, California, 1991.
27. K. Fukunaga, Introduction to Statistical Pattern Recognition (Second Edition), Academic Press, New York, 1990.
28. Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison Wesley, Reading, Massachusetts, 1989.
29. Satoshi Watanabe, Pattern Recognition, Human and Mechanical, John Wiley & Sons, New York, 1985.
30. T.Y. Young and K.S. Fu, Handbook of Pattern Recognition and Image Processing, Academic Press, Orlando, Florida, 1986.
31. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and regression trees, Wadsworth, California, 1984.
32. P.A. Devijver and J. Kittler, Pattern Recognition, a Statistical Approach, Prentice Hall, Englewood Cliffs, London, 1982.
33. R.C. Gonzalez and M.G. Thomason, Syntactic pattern recognition - An introduction, Addison-Wesley, Reading, 1982.
34. J. Sklanski and G.N. Wassel, Pattern Classifiers and Trainable Machines, Springer, New York, 1981.
35. R.O. Duda and P.E. Hart, Pattern classification and scene analysis, John Wiley & Sons, New York, 1973.
36. R.O. Duda, P.E. Hart, and D. Stork, Pattern classification, 2nd edition, John Wiley & Sons, 2001.

37. P. Dayan, L.F. Abbott, Theoretical Neuroscience, Computational and Mathematical Modeling of Neural Systems , MIT Press, December 2001.
38. U. Seiffert, L.C. Jain (editors), Self-Organizing Neural Networks: Recent Advances and Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag, November 2001.
39. W. Maass and C. M. Bishop, editors, Pulsed Neural Networks, MIT Press, Cambridge, 1999.
40. S. Amari, N. Kasabov, Brain-like computing and intelligent information systems, Springer Verlag, Berlin, 1998.
41. G. B. Orr, K-R. Müller (editors), Neural Networks: Tricks of the Trade, Springer-Verlag Berlin Heildeberg, 1998.
42. T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1995, 1997.
43. C. M. Bishop, editor, Neural Networks and Machine Learning 1997 NATO Advanced Study Institute, Springer 1998.
44. P. Smolensky, M. C. Mozer, and D. E. Rumelhart, Mathematical Perspectives on Neural Networks, Lawrence Erlbaum Associates, Inc. Mahwah, New Yersey, 1996.
45. Y. Bengio, Neural networks for speech and sequence recognition, International Thomson Publishing, London, 1995.
46. LiMin Fu, Neural Networks in Computer Intelligence, McGraw-Hill, Inc., New York, NY, 1994.
47. S. Haykin, Neural Networks, A Comprehensive Foundation, Macmillan, New York, NY, 1994.
48. S.Y. Kung, Digital Neural Networks, Prentice Hall, Englewood Cliffs, NJ, 1993.
49. Stephen I. Gallant, Neural Network Learning and Expert systems, Massachusetts Inst. of Technology, Cambridge, Massachusetts, 1993.

50. A. Cichocki and R. Unbehauen, Neural Networks for Optimization and Signal Processing, John Wiley & Sons, New York, 1993.
51. C. H. Chen, L. F. Pau, P. S. P. Wang, Handbook of Pattern Recognition and Computer Vision, World Scientific, Singapore, 1993.
52. B. Kosko, Neural networks for signal processing, Prentice-Hall, Englewood Cliffs, 1992.
53. J.M. Zurada, Artificial Neural Systems, West Publishing, St. Paul, MN, 1992.
54. B. Muller and J. Reinhardt, Neural networks, an introduction, Springer-Verlag, Berlin, 1991.
55. John Hertz, Anders Krogh, and Richard G. Palmer, Introduction to the Theory of Neural Computation, Addison Wesley Publ. Comp., Redwood City ,CA, 1991.
56. J. Diederich, Artificial neural networks - Concept learning, IEEE Computer Society Press, Los Alamitos, 1990.
57. P.D. Wasserman, Neural Computing, theory and practice, Van Nostrand Reinhold, New York, 1989.
58. I. Aleksander, Neural Computing Architectures, North Oxford Academic, London, 1989.
59. S. Grossberg, The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rythm, Elsevier/North Holland, Amsterdam, 1987.
60. S. Grossberg, The Adaptive Brain II: Vision, Speech, Language and Motor Control, Elsevier/North Holland, Amsterdam, 1987.
61. B. V. K. Vijaya Kumar, A. Mahalanobis, Richard Juday, Correlation Pattern Recognition, Cambridge University Press, 2005.
62. B. V. K. V. Kumar, "Tutorial survey of composite filter designs for optical correlators," Appl. Opt. 31, 4773- (1992)

63. C. F. Olson and D. P. Huttenlocher, "Automatic Target Recognition by Matching Oriented Edge Pixels," IEEE Trans. On Image Processing, 6(1), pp. 103-113, January 1997.
64. C. E. Daniell, D. H. Kemsley, W. P. Lincoln, W. A. Tackett and G. A. Baraghimian, "Artificial neural networks for automatic target recognition," Opt. Eng., 31(12), , pp. 2521–2531 December 1992.
65. J., A. O'Sullivan, M., D. DeVore, V. Kedia and M. I. Miller, "SAR ATR Performance Using A Conditionally Gaussian Model," IEEE Trans. on Aerospace and Electronic Systems, 37(1), pp. 91-108, January 2001.
66. S. G. Sun, D. M. Kwak, W. B. Jang, D. J. Kim, "Small Target Detection using Center-surround Difference with Locally Adaptive Threshold," in Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, ISPA 2005, September 2005, pp. 402-407.
67. C. F. Olson, D. P. Huttenlocher and D. M. Doria, "Recognition by matching with edge location and orientation," in Proceedings of the ARPA Image Understanding Workshop, 1996, pp. 1167-1174.
68. F. Sadjadi, "Object recognition using coding schemes," Opt. Eng., 31(12), pp. 2580-2583, December 1992.
69. J. G. Verly, R. L. Delanoy and D. E. Dudgeon, "Model-based system for automatic target recognition from forward-looking laser-radar imagery," Opt. Eng., 31(12), pp. 2540-2552, December 1992.
70. B. Bhanu, and J. Ahn, "A system for model-based recognition of articulated objects," in Proceedings of the International Conference on Pattern Recognition, 1998, 1812-1815.

71. S. Z. Der, Q. Zheng, R. Chellappa, B. Redman and H. Mahmoud, "View based recognition of military vehicles in LADAR imagery using CAD model matching," in Image Recognition and Classification, Algorithms, Systems and Applications, B. Javidi, ed., (Marcel Dekker, Inc., 2002), pp. 151-187.

72. J. Starch, R. Sharma and S. Shaw, "A unified approach to feature extraction for model based ATR," Proc. SPIE **2757**, 294-305, 1997.

73. D. Casasent and R. Shenoy, "Feature space trajectory for distorted object classification and pose estimation in SAR," Opt. Eng., 36(10), pp. 2719-2728, 1997.

74. D. P. Kottke, J. Fwu and K. Brown, "Hidden Markov Modelling for Automatic Target Recognition," Conference Record of the Thirty-First Asilomar Conference on Signals, Systems, & Computers, vol. 1, November 2-5, 1997, pp. 859-863.

75. S. A. Rizvi and N. M. Nasrabadi, "Automatic target recognition of cluttered FLIR imagery using multistage feature extraction and feature repair," Proc. SPIE **5015**, 1-10, March 2003.

76. L. A. Chan, S. Z. Der and N. M. Nasrabadi, "Neural based target detectors for multi-band infrared imagery," In B. Javidi (Ed.), Image Recognition and Classification, Algorithms, Systems and Applications, New York: Marcel Dekker, Inc., 2002, 1—36.

77. D. Torreiri, "A linear transform that simplifies and improves neural network classifiers," In Proceedings of International Conference on Neural Networks, Vol. 3, 1996, 1738—1743.

78. J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Ann. Stat. 29, 1189–1232, 2001.

79. H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support vector regression machines," Adv. Neural Inf. Process. Syst. 9, pp. 155–161, 1997.

80. H. C. Chiang, R.L. Moses and W.W. Irving, "Performance estimation of model-based automatic target recognition using attributed scattering center features," Proceedings of the International Conference on Image Analysis and Processing, Venice, Italy, 1999, pp. 303-308.

81. S. A. Rizvi and N. M. Nasrabadi, "Fusion Techniques for Automatic Target Recognition," 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03), 2003, pp. 27- 32.

82. D. Casasent and Y.C. Wang, "Automatic Target Recognition Using New Support Vector Machine," Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, IJCNN 2005, 31 July – 4 Aug., vol. 1, pp. 84-89.

83. R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification," Wiley-Interscience; 2nd edition, October 2000.

84. B. V. K. Vijaya Kumar, "Tutorial survey of composite filter designs for optical correlators," Applied Optics, 31 (1992), pp. 4773—4801.

85. A. Mahalanobis, B. V. K. Vijaya Kumar, S. R. F. Sims and J. Epperson, "Unconstrained correlation filters," Applied Optics, 33 (1994), pp. 3751—3759.

86. X. Huo, M. Elad, A. G. Flesia, R. R. Muise, S. R. Stanfill, J.Friedman, B. Popescu, J. Chen, A. Mahalanobis and D. L. Donoho, "Optimal reduced-rank quadratic classifiers using the Fukunaga–Koontz transform with applications to automated target recognition," Automatic Target Recognition XIII, F. A. Sadjadi, ed., Proc. SPIE 5094, 2003, pp. 59–72.

87. S. R. F. Sims and A. Mahalanobis, "Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery," Optical Engineering, 43(8), August 2004, pp. 1705-1711.

88. A. Mahalanobis, R. R. Muise and S. R. Stanfill, "Quadratic Correlation Filter Design Methodology for Target Detection and Surveillance Applications," Applied Optics, 43(27), 2004, pp. 5198-5205.

89. R. Muise, A. Mahalanobis, R. Mohapatra, X. Li, D. Han and W. Mikhael, "Constrained Quadratic Correlation Filters for Target Detection," Applied Optics, 43(2), 2004, pp. 304-314.

90. K. R. Rao and P. Yip, “Discrete Cosine Transform: Algorithms, Advantages, Applications”, Academic Press, Boston, 1990.

91. E. Feig and S. Winograd, “On the multiplicative complexity of discrete cosine transforms,” IEEE Transactions on Information Theory, vol. 38, July 1992, pp. 1387–1391.

92. H. R. Wu and Z. Man, "Comments on Fast algorithms and implementation of 2-D discrete cosine transform”, IEEE Transactions on Circuits and Systems for Video Technology, Volume: 8, Issue: 2, Apr 1998, pp. 128-129.

93. C. Chen, B. Liu and J. Yang, "Direct recursive structures for computing radix-r two-dimensional DCT/IDCT/DST/IDST," IEEE Transactions on Circuits and Systems I: Regular Papers, Volume: 51, Issue: 10, Oct. 2004, pp. 2017- 2030.

94. A. K. Abed-Meraim, A. Chkeif, Y. Hua, “Fast Orthogonal PAST Algorithm,” IEEE Signal Processing Letters, no. 7, pp. 60-62, March 2000.

95. S. Attallah, K. Abed-Meraim, “Fast algorithms for Subspace Tracking,” IEEE Signal Processing Letters, vol. 8, pp. 203-206, July 2001.

96. P.F. Baldi, K. Hornik, “Learning in Linear Neural Networks: A Survey,” IEEE Transactions on Neural Networks, vol. 6, no.4, pp. 837-858, July 1995.

97. S. Bannour, M.R. Azimi-Sadjadi, “Principal Component Extraction Using Recursive Least Squares Learning,” IEEE Transactions on Neural Networks, vol. 6, no.2, pp. 457-469, March 1995.

98. B. Champagne, “Adaptive eigendecomposition of data covariance matrices based on first order perturbations”, IEEE Transactions on Signal Processing, Vol. 42, No. 10, October 1994.

99. B. Champagne, Q. Liu, “Plane Rotation-Based EVD Updating Schemes for Efficient Subspace Tracking,” IEEE Transactions on Signal Processing, vol. 46, no. 7, July 1998.

100. C. Chatterjee, Z. Kang, V.P. Roychowdhury, “Algorithms for Accelerated Convergence of Adaptive PCA,” IEEE Transactions on Neural Networks, vol.11, no. 2, pp. 338-355, March 2000.

101. C. Chatterjee, V.P. Roychowdhury, “Adaptive Algorithms for Eigen-Decomposition and Their Applications in CDMA Communication Systems,” Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, vol.2, pp. 1575-1580, 1997.

102. C. Chatterjee, V. P. Roychowdhury, “An Adaptive Stochastic Approximation Algorithm for Simultaneous Diagonalization of Matrix Sequences With Applications,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 3, March 1997.

103. Y. Chauvin, “Principal component analysis by gradient descent on a constrained linear Hebbian cell,” Proceedings of the International Joint Conference on Neural Networks (IJCNN), vol.1, pp. 373-380, 1989

104. T. Chonavel, B. Champagne, C. Riou, “Fast adaptive eigenvalue decomposition: a maximum likelihood approach”, Signal Processing, 83 (2003) 307 – 324.

105. P. Comon, G.H. Golub, “Tracking a few extreme singular values and vectors in signal processing,” Proceedings of the IEEE, vol. 78, no. 8, pp. 1327-1343, August 1990.

106. C. E. Davila, “Blind Adaptive Estimation of KLT Basis Vectors,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, June 5-9, 2000.

107. R.D. DeGroat, “Noniterative Subspace Tracking”, IEEE Transactions on Signal Processing, vol. 40, pp. 571-577, March 1992.

108. R.D. DeGroat, E.M. Dowling, D.A. Linebarger, “Subspace tracking”, The Digital Signal Processing Handbook, eds. V.K. Madisetti, D.B. Williams, CRC Press, Boca Raton, FL, 1998.

109. R. D. DeGroat, R. A. Roberts, “SVD Update Algorithm and Spectral Estimation Application,” Proceedings of the 19th Asilomar Conference on Circuits, Systems and Computers, pp. 601-605, 1985.

110. R. D. DeGroat, R. A. Roberts, “Tracking Non-Stationary Narrowband Signals with Reduced Rank Updating,” Proceedings of the 20th Asilomar Conference on Circuits, Systems and Computers, pp. 447-451, 1986.

111. R. D. DeGroat, R. A. Roberts, “An Improved, Highly Parallel Rank-one Eigenvector Update Method with Signal Processing Application,” Proc. SPIE Int. Soc. Opt. Eng., vol. 696, pp. 62-70, 1986.

112. R. D. DeGroat, R. A. Roberts, “Highly Parallel Eigenvector Update Methods with Applications to Signal Processing,” Proceedings of the 21st Asilomar Conference on Circuits, Systems and Computers, pp. 564-568, 1987.

113. J.P. Delmas, J.F. Cardoso, “Performance Analysis of an Adaptive Algorithm for Tracking Dominant Subspaces,” IEEE Transactions on Signal Processing, vol. 46, no.11, pp. 3045-3057, November 1998.

114. Z. Fu and E. M. Dowling, “Conjugate Gradient Eigenstructure tracking for adaptive spectral estimation,” IEEE Trans. Signal Processing, vol. 43, pp. 1151–1160, 1995.

115. Z. Fu and E. M. Dowling, “Conjugate Gradient Projection Subspace Tracking,” IEEE Transactions on Signal Processing, vol. 45, no. 6, pp. 1664-1668, June 1997.

116. D. R. Fuhmzann, A. Srivastava, H. Moon, “Subspace Tracking via Rigid Body Dynamics,” 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing (SSAP), p. 578, 1996.

117. S. Haykin, “Neural Networks – A Comprehensive Foundation,” Macmillan, New York, 1994.

118. Y. Lau, Z. M. Hussian, R. Harris, “A Time-Varying Convergence Parameter for the LMS Algorithm in the Presence of White Gaussian Noise”, Australian Telecommunications, Networks and Applications Conference (ATNAC), 2003.

119. J. Lee, K. Bae, “Numerically Stable Fast Sequential Calculation for Projection Approximation Subspace Tracking,” Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS), vol.3, 484-487, 1999.

120. A. Mahalanobis, R. R. Muise, S. R. Stanfill, “Quadratic Correlation Filter Design Methodology for Target Detection and Surveillance Applications”, Applied Optics, Volume 43, Issue 27, 5198-5205, September 2004.

121. G. Mathew, V. U. Reddy, and S. Dasgupta, “Adaptive estimation of eigensubspace,” IEEE Trans. Signal Processing, vol. 43, pp. 401–411, 1995.

122. T. McWhorter, “Fast Rank-Adaptive Beamforming,” Proceedings of the 2000 IEEE Sensor Array and Multichannel Signal Processing Workshop. SAM 2000, pp. 63-67.

123. Y. Miao, “Comments on Principal Component Extraction Using Recursive Least Squares Learning,” IEEE Transactions on Neural Networks, vol. 7, no.4, pp. 1052, July 1996.

124. Y. Miao, Y. Hua, “Fast Subspace Tracking and Neural Network Learning by a Novel Information Criterion,” IEEE Transactions on Signal Processing, vol. 46, no.7, pp. 1967-1979, July 1998.

125. C. Michalke, M. Stege, F. Schäfer, G. Fettweis, “Efficient tracking of eigenspaces and its application to eigenbeamforming,” IEEE PIMRC’03, Band 3, S. 2847-2851. Beijing, China, 7.-10. S. 2003.

126. C.B. Moller, G.W. Stewart, “An Algorithm for Generalized Matrix Eigenvalue Problems”, SIAM J. Numer. Anal., Vol. 10, No. 2, April 1973.

127. W. Mikhael and F. Wu, “A fast block FIR adaptive digital filtering algorithm with individual adaptation of parameters”, IEEE Transactions on Circuits and Systems, Vol. 36, No. 1, January 1989.

128. R. Muise, A. Mahalanobis, R. Mohapatra, X. Li, D. Han, and W. Mikhael, “Constrained quadratic correlation filters for target detection”, Applied Optics, Vol. 43, No. 2, January 2004.

129. E. Oja, “Neural Networks, Principal Components, and Subspaces,” International Journal of Neural Systems, vol. 1, pp. 61-68, April 1989.

130. E. Oja, H. Ogawa, J. Wangviwattana, “Principal Component Analysis by Homogeneous Neural Networks, Part I: The Weighted Subspace Criterion,” IEICE Trans. Inf. & Syst., vol. E75-D, pp. 366-375, May 1992.

131. D. J. Rabideau, “Fast, rank adaptive subspace tracking and applications,” IEEE Trans. Acoust. , Speech, Signal Processing, vol. 44, pp. 2229–2244, September 1996.

132. E. C. Real, D. W. Tufts, J. W. Cooley, “Two Algorithms for Fast Approximate Subspace Tracking,” IEEE Transactions on Signal Processing, vol. 47, no. 7, July 1999.

133. P.A. Regalia, “An Adaptive Unit Norm Filter with Application to Signal Analysis and Karhunen-Loeve Transformations,” IEEE Transactions on Circuits and Systems, vol. 37, pp. 646-649, May 1990.

134. D. Reynolds, X. Wang, “Adaptive Group-Blind Multiuser Detection Based on a New Subspace Tracking Algorithm,” IEEE Transactions on Communications, vol. 49, no.7, pp. 1135-1141, July 2001.

135. C. Riou , T. Chonavel, “Fast adaptive eigenvalue decomposition: a maximum likelihood approach”, Proceedings of the ICASSP ’97, pp. 3565 – 3568, 1997.

136. C. Riou , T. Chonavel, P.Y. Cochet, “Adaptive Subspace Estimation: Application to Moving Sources Localization and Blind Channel Identification,” Proceedings of the ICASSP ’96, pp. 1649 – 1652, 1996.

137. T. Tanaka, “Generalized weighted rules for principal components tracking,”, IEEE Trans. Signal Processing, vol.53, no.4, pp.1243-1253, April 2005.

138. T. Tanaka, “Generalized subspace rules for on-line PCA and their application in signal and image compression,” in Proc. of 2004 IEEE International Conference on Image Processing (ICIP 2004), vol.II, pp.1895-1898, Singapore, Oct. 2004.

139. D. S. Watkins, “Product Eigenvalue Problems”, SIAM Review, Vol. 47, No. 1, pp. 3-40, March 2005.

140. L. Xu, “Least Mean Square Error Reconstruction Principle for Self-organizing Neural Nets,” Neural Networks, vol. 6, pp. 627-648, 1993.

141. B. Yang, “Subspace Tracking based on the Projection Approach and the Least Squares Method,” Proceedings of the ICASSP ’93, pp. 145 – 148, 1993.

142. B. Yang, “Projection Approximation Subspace Tracking,” IEEE Transactions on Signal Processing, vol. 43, no.1, pp. 95-107, January 1995

143. X. Yang, T. K. Sarkar, and E. Arvas, “A survey of conjugate gradient algorithms for solution of extreme eigen-problems of a symmetric matrix,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 37, pp. 1550–1556, 1989.

144. Y.F. Yang, M. Kaveh, “Adaptive Eigensubspace algorithms for direction or frequency estimation and tracking,” IEEE Transactions on Acoustics and Speech Signal Processing, vol. 36, no. 6, pp. 241- 251, June 1988.

145. K. Yu, “Recursive updating the eigenvalue decomposition of a covariance matrix,” IEEE Transactions on Signal Processing, Vol. 39, No. 5, May 1991.

146. G.H. Golub, C. F. Van Loan, “Matrix Computations,” The Johns Hopkins University Press, 3 edition (October 15, 1996)

147. R. R. Muise, “Quadratic Filters for Automatic Pattern Recognition,” Ph.D. Dissertation, University of Central Florida, Orlando, FL, USA, 2003.

148. Sayood, K., Introduction to Data Compression. Morgan Kaufmann Publishers, 2000.

149. Clarke, R.J., Transform Coding of Images. Academic Press, 1985.

150. Bhaskaran, V., Konstantinides, K., Image and Video Compression Standards – Algorithms and Architectures, Kluwer Academic Publishers, 1997.