A Novel Adaptive Eigendecomposition Technique with Application to Automatic Target Recognition


* School of Electrical Engg. & Comp. Sc., Univ. of Central Florida, Orlando, FL 32816, USA
** Embry-Riddle Aeronautical Univ., 600 S. Clyde Morris Blvd, Daytona Beach, FL 32114, USA
*** Lockheed Martin, MFC, 5600 Sand Lake Road, Orlando, FL 32819-8907, USA

Abstract: - Among many methods for Automatic Target Recognition (ATR), Quadratic Correlation Filters (QCF) have become popular. Recently, the Rayleigh Quotient Quadratic Correlation filters (RQQCF) technique was proposed to optimize QCF’s for ATR by explicitly maximizing a class separation metric. The method requires the Eigenvalue Decomposition (EVD) of a large matrix, which is obtained using the autocorrelation matrices of target and clutter training images. Often, in practice, when few new targets and clutter need to be incorporated, the EVD is perturbed. It is desirable to have methods that eliminate the need to perform an EVD every time new data is to be added. In this paper, a novel method is proposed to obtain the new EVD adaptively, starting from the initial EVD, leading to significant computational savings. Sample results using synthetic datasets confirm the excellent properties of the technique

Keywords: - Automatic Target Recognition, Adaptive Eigendecomposition, Quadratic Correlation Filters.

1 Introduction
There are many methods in the literature for ATR. Some involve applying tools such as neural networks, [1], Support Vector Machines (SVM), [2], and decision trees, [3], to classify features extracted from images. Others segment the objects of interest and try to model them, [4], [5]. Recently, QCF’s have gained popularity because of their inherent shift-invariance, and also because they do not require feature extraction or segmentation. In [6], a novel method is proposed for optimizing QCF’s for target detection in Infrared (IR) images. The task of recognizing and classifying targets and background clutter is viewed as a two-class pattern recognition problem. The class-separation metric is formulated as a Rayleigh quotient that is maximized by the QCF solution. As a result, the means of the two classes are well separated while simultaneously ensuring that the variance of each class is small. In the training mode, a set of target and clutter sub-images, referred to as chips, are extracted from IR images. Autocorrelation matrices for target and clutter are computed. These matrices are appropriately combined to obtain a non-symmetric matrix A, as shown in Section 2. The EVD of A is performed. The dominant eigenvalues of A are \( \lambda_{ti} \), corresponding to target, and \( \lambda_{ci} \), corresponding to clutter, \( i=1 \text{ to } k \). The QCF coefficients are the set of eigenvectors \( w_{ti} \) and \( w_{ci} \), corresponding to \( \lambda_{ti} \) and \( \lambda_{ci} \). In the running mode, the QCF is correlated with a given IR image. Based on the response obtained, it is able to recognise those targets that were used in training. Often, in practice, the system is required to incorporate few new targets relative to the typically large number of data points used in training. Thus \( w_{ti} \) and \( w_{ci} \) become \( \hat{w}_{ti} \) and \( \hat{w}_{ci} \), which are in the vicinity of \( w_{ti} \) and \( w_{ci} \). In such cases, the target and/or clutter set has to be updated, and the QCF has to be retrained. This involves the inversion and EVD of large matrices which is computationally expensive. Therefore, it is desirable to have an adaptive EVD algorithm that reduces the computations. There are several contributions in the literature that address the problem of adaptive EVD. Some of the more recent contributions are given in [7]-[9]. These are typically
formulated for the adaptive EVD of symmetric matrices. In this paper, an efficient adaptive technique called the Optimal Adaptive Eigenvalue Decomposition (OAEVD) technique, is proposed. It avoids matrix inversion and direct EVD, and instead utilizes the old EVD to search for the new EVD. Thus, it provides substantial computational savings. The OAEVD also adaptively updates any particular set of eigenvalues and corresponding eigenvectors of interest. In our application, these are the dominant eigenvalues and corresponding eigenvectors. The proposed OAEVD technique lends itself to parallel implementations because it can track the dominant eigenpairs independent of each other. Sample results obtained confirm the excellent properties of the proposed technique. The remainder of the paper is organized as follows: Section 2 gives a brief summary of [6] before describing the proposed technique, Section 3 contains sample simulation results, and Section 4 contains the conclusions.

2 Proposed Technique

A brief summary of [6] is given first as the OAEVD technique, Section 3 contains sample simulation results, and Section 4 contains the conclusions.

The OAEVD technique is based on the RQQCF technique. In the RQQCF technique, M target and M clutter training chips are obtained from IR imagery. Each chip, having dimensions $\sqrt{n} \times \sqrt{n}$, is converted into a 1-D vector of dimensions $n \times 1$ by concatenating its columns. Target and clutter training sets of size $n \times M$ each, are obtained by placing the respective vectors in matrices. The $n \times n$ autocorrelation matrices of the target and clutter sets, $R_t$ and $R_c$, are computed, and used to obtain a matrix $A$ given by,

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

As a result of (1), the eigenvalues of $A$ vary from $-1$ to $+1$. The dominant eigenvalues for clutter, $\lambda_{cl}$, are close to or equal to $-1$ and those for targets, $\lambda_{ti}$, are close to or equal to $+1$. The RQQCF coefficients, $w_{ci}$ and $w_{ti}$, are mapped to the corresponding eigenvalues. To identify a data point as target or clutter, the sum of the absolute value of the $k$ inner products of the data point with $w_{ij}$ and $w_{cl}$, $p_t$ and $p_c$, are calculated. If $p_c > p_t$, the data point is identified as a target. Otherwise, it is identified as clutter. When new data points have to be incorporated, $R_c$ and/or $R_t$ change, and $A$ has to be recomputed. In addition, the EVD of $A$ has to be performed again. Hence an efficient adaptive approach is proposed. In the following formulation, quantities in lowercase with an underscore are vectors, and quantities in upper case are matrices. The first step in the adaptive formulation is to identify a “Cost Function” to be minimized when new data is incorporated. According to (1), the following is chosen as the “Error Signal” at the $j^{th}$ iteration:

$$e(j) = [R_x - R_y - \lambda_i(j)(R_x + R_y)]w_{ij}(j)$$ (2)

where, $e(j)$ is $n \times 1$, $\lambda_i$ and $w_{ij}$ are the $i^{th}$ Eigenvalue and the corresponding Eigenvector respectively. The energy in the error signal, $e^T(j)e(j)$, is the cost function to be minimized. The error signal contains both $\lambda_i$ and $w_{ij}$. It can be easily seen that the error surface is multi-modal. In addition, in practice, the new $\lambda_i$’s and $w_{ij}$’s, $\dot{\lambda}_i$’s and $\dot{w}_i$’s, are in the vicinity of the old ones. Thus, in the adaptive algorithm, the $\lambda_i$’s and $w_{ij}$’s are updated in an alternating manner. In each iteration, $\dot{\lambda}_i(w_{ij})$ is adjusted while $w_{ij}(\lambda_i)$ is kept constant. To obtain $\lambda_i(j+1)$ ($w_{ij}(j+1)$), the most recent updated $w_{ij}$’s ($\lambda_i$’s) and $\lambda_i$’s ($w_{ij}$’s) are used. Again, since the $\dot{\lambda}_i$’s and $\dot{w}_i$’s are close to the old ones, only the first order terms are retained in the Taylor series expansion for $e(j+1)$ in terms of $e(j)$. This yields,

$$e(j+1) = e(j) + \frac{\partial e(j)}{\partial \lambda_i(j)} \Delta \lambda_i(j)$$ (3)

where, $l = 1, \ldots, n$. Writing (3) for $l = 1, \ldots, n$:

$$e(j+1) = e(j) + (S_1 - \lambda_i(j)S_2)\Delta \lambda_i(j) - S_2 \Delta w_{ij}(j)$$ (4)

where, $S_1 = R_x - R_y$, $S_2 = R_x + R_y$. Now, to obtain $\lambda_i(j+1)$, the most recent $w_{ij}$ is used. Therefore, (4) becomes

$$e(j+1) = e(j) - S_2 \Delta w_{ij}(j) \lambda_i(j)$$ (5)
To obtain $w_i(j+1)$, the most recent $\lambda_i$ is employed. Thus, (4) becomes

$$e(j+1) \approx e(j) + (S_1 - \lambda_i(j)S_2)\Delta w_i(j)$$

(6)

In the steepest descent adaptation, $\Delta w_i(j)$ and $\Delta \lambda_i(j)$ are proportional to the negative gradient of the cost function in the $j^{th}$ iteration, $e^T(j)e(j)$, with respect to $w_i(j)$ and $\lambda_i(j)$, respectively. Therefore,

$$\Delta \lambda_i(j) = -k_{\lambda_i} \frac{\partial(e^T(j)e(j))}{\partial \lambda_i(j)}$$

$$= -2k_{\lambda_i} w_i^T(j)S_2(\lambda_i(j)S_2 - S_1)w_i(j)$$

(7)

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

(8)

where, $k_{\lambda_i}$ is a real scalar, and $[MU]$ is a diagonal matrix with real, positive entries. From (5), (6), (7), and (8), it can be shown that the update equations are:

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

$$\Delta w_i(j) = -[R]^{-1}q(j)$$

where,

$$V^T(j) = S_2 w_i(j)(\lambda_i(j)w_i^T(j)S_2 - w_i^T(j)S_1)w_i(j)$$

$$q(j) = Se(j)$$

and $[R]_j = [S]^2$.

The above algorithm avoids the EVD required for the updated $A$. As a further computational reduction in the proposed adaptive solution, $[R]_j^{-1}$ is approximated by a diagonal matrix $B$ containing the reciprocals of $R$'s diagonal elements, $r_{ii}(i = 1 \text{ to } n)$. Additionally, a convergence factor $\mu$ is introduced in the update equations to ensure reliable convergence. Thus

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

$$\Delta w_i(j) = -\mu B q(j)$$

3 Simulation results

The adaptive algorithm was tested using various 1D and 2D synthetic datasets. Due to space constraints, only a sample result for one of the 2D datasets is presented. The algorithm was found to be successful in extensive simulations for the other datasets also.

Dataset: 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. Fig. 1 and Fig. 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.

The matrix $A$ is computed according to (1). The EVD of $A$ is performed to obtain the initial $\lambda_i$’s and $w_i$’s. Then, the new data points are added and the adaptive algorithm is used to track the changes in the $\lambda_i$’s and $w_i$’s. For the sake of evaluating the new algorithm, the $\hat{\lambda}_i$’s and $\hat{w}_i$’s obtained from the OAEVD algorithm are compared with the exact values $\lambda_{ie}$’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 $\varepsilon$. All simulations are performed using MATLAB 7. $\mu$ and $\varepsilon$ used are 0.1 and 1e-25, respectively.

Fig 1 Sample target chip
Sample Result: 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 $\hat{w}_i$'s. Five dominant target and Five dominant clutter eigenpairs are computed. The result is summarized below.

$$\hat{\lambda}_i = -0.9772, -0.9651, -0.8899, -0.8774, -0.8317, 0.8555, 0.9459, 0.9482, 0.9710, 0.9997$$

$$\hat{\lambda}_{te} = -0.9768, -0.9340, -0.8932, -0.8617, -0.6847, 0.8531, 0.9444, 0.9463, 0.9719, 0.9997$$

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

$\text{iter} = 132, 237, 107, 156, 261, 65, 123, 172, 12$

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, $w_i$, $\hat{w}_i$ & $w_{te}$ are not given because of space constraints. Fig. 3 and Fig. 4 show plots of the absolute value of the inner product of the new data point that was added, with $\hat{w}_i$ and $w_{te}$, respectively. There is a good match between the two plots. Fig. 5 and Fig. 6 show the corresponding plots for a clutter point randomly selected from the training set.

4 Conclusion

A novel OAEVD algorithm for RQQCF based ATR is proposed. The sample results presented above show that the algorithm is able to track changes in the EVD when few new data points are added. It uses the old EVD to search for the new EVD thereby eliminating the need to perform matrix inversion and EVD. This results in significant computational savings. 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. We note that 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. The proposed algorithm is dependent on $\mu$ and $\varepsilon$. Further work is underway for the optimal selection of these parameters. Work is also in progress to apply the OAEVD to an Infrared (IR) dataset, which will be reported in the near future.

References:


Fig 3 Absolute value of inner product of the new target vector with \(^{\hat{W}}_{ij}\) 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).

Fig 4 Absolute value of inner product of the new target vector with \(^{\hat{W}}_{ie}\) 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).

Fig. 5 Absolute value of inner product of a clutter vector with \(^{\hat{W}}_{ij}\) versus \(i\), the index of the \(^{\hat{W}}_{ij}\)’s. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).

Fig. 6 Absolute value of inner product of a clutter vector with \(^{\hat{W}}_{ie}\) versus \(i\), the index of the \(^{\hat{W}}_{ie}\)’s. The first five Eigenvectors correspond to clutter (eigenvalues close to -1) and the next five correspond to target (eigenvalues close to +1).