Constrained quadratic correlation filters for target detection

Robert Muise, Abhijit Mahalanobis, Ram Mohapatra, Xin Li, Deguang Han, and Wasfy Mikhael

A method for designing and implementing quadratic correlation filters (QCFs) for shift-invariant target detection in imagery is presented. The QCFs are quadratic classifiers that operate directly on the image data without feature extraction or segmentation. In this sense the QCFs retain the main advantages of conventional linear correlation filters while offering significant improvements in other respects. Not only is more processing required for detection of peaks in the outputs of multiple linear filters but choosing the most suitable among them is an error-prone task. All channels in a QCF work together to optimize the same performance metric and to produce a combined output that leads to considerable simplification of the postprocessing scheme. The QCFs that are developed involve hard constraints on the output of the filter. Inasmuch as this design methodology is indicative of the synthetic discriminant function (SDF) approach for linear filters, the filters that we develop here are referred to as quadratic SDFs (QSDFs). Two methods for designing QSDFs are presented, an efficient architecture for achieving them is discussed, and results from the Moving and Stationary Target Acquisition and Recognition synthetic aperture radar data set are presented.

OCIS codes: 120.2440, 100.0100, 100.5010, 100.6740.

1. Introduction

The problems of automatic target recognition (ATR) have been researched in several scientific and engineering disciplines. Automatic target detection (ATD) is the problem of determining whether an image contains an interesting object and, if it does, estimating the pixel location of that object in the image. ATR is the problem of analyzing an interesting object identified by a detection algorithm and assigning a class label to that object. Although the techniques described in this paper are applicable to any imaging sensor, we present a case study in which we use synthetic-aperture radar (SAR) image data. Also, these techniques are applicable to any patterns in one-dimensional signals or two-dimensional images. However, in our presentation of the theory and the results we shall refer to ATD and ATR in image data. Considerable research on ATD and ATR has been done over the past three decades. Popular methods for achieving this goal include statistical techniques based on feature classification,[1,2] model-based techniques,[3,4] neural networks,[5,6] and maximum-likelihood algorithms.[7] Generally speaking, the potential targets are segmented from the background; boundary edges and internal features are measured and fitted to either statistical or structural models. These methods work well when there are sufficient numbers of pixels on the targets and the contrast boundaries are well defined. In general, however, the performance of most segmentation-based methods suffer considerably when the targets exhibit poor contrast relative to the background or when parts of the target are either obscured or in close proximity to clutter. In such cases segmentation is a difficult task.

In this paper the focus is on target-detection techniques based on signal-processing methods that do not require segmentation. The leading approach is based on the use of correlation filters.[8,9] Perhaps the earliest and best known among these filters is the matched spatial filter (MSF).[10] Although the MSF is known to be optimum for the detection of a single (deterministic) pattern in the presence of additive white Gaussian noise, it is hardly suitable for robust

R. Muise (robert.r.muise@lmco.com) and A. Mahalanobis are with Lockheed Martin, MFC, MP 450, 5600 Sand Lake Road, Orlando, Florida 32819. R. Mohapatra, X. Li, and D. Han are with the Department of Mathematics, University of Central Florida, Orlando, Florida 32816-1364. W. Mikhael is with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816-2450.

Received 12 May 2003; revised manuscript received 29 July 2003; accepted 25 August 2003.

0003-6935/04/020304-11$15.00/0

detection and recognition of targets in real-world images. To add robustness and tolerance to distortion to the detection process, consider both the data and the pattern as being stochastic processes. This no longer fits the model in which the MSF is optimal, and a different methodology is needed for defining correlators for the detection problem. As a result, many approaches to the design of filters for correlators that exhibit robust behavior in the presence of clutter and of wide variations in the targets' signatures have been proposed. An extension of the linear techniques to the realm of quadratic correlation filters (QCFs), as well as an efficient implementation algorithm, was presented by Mahalanobis *et al.*[11] In that paper a general framework for quadratic filters for the ATD–ATR problem was presented from a fundamental mathematical model, with the types of objective function and design techniques from the linear filter design literature that had not yet been pursued systematically and an implementation framework that utilizes a bank of linear filters to compute the quadratic filter output. The novelty of that paper is a description of the design of two new QCFs for an ATD, one based on the optimal separation of information of the two classes given by the Fukanaga–Koontz transform and the other based on an optimal separation criterion that uses a modified Fisher ratio and results in the form of a Rayleigh quotient. To pursue further the quadratic model for correlation filtering, it makes sense to follow the historical development of the variety of linear design techniques that have been developed over the years. Synthetic discriminant function (SDF) filters[12,13] were developed to improve the distortion tolerance of the MSF algorithm. When target distortions (rotations and viewing angle changes) cannot be accounted for analytically, one assumes that the training data are reasonably representative of the anticipated distortions. With this assumption, defining hard constraints on the output of the filter for the training data will increase the distortion tolerance of the designed filter. This is the concept behind the synthetic approach to discriminant function design. In this paper, design algorithms for QCFs are presented that involve hard constraints on the output of the filter; therefore these methods generalize the linear SDF algorithms.[12,13]

The remainder of this paper is organized as follows: In Section 2 we review the concepts of designing linear synthetic discriminant functions for object detection and present design techniques for two quadratic synthetic discriminant function (QSDF) detection filters. In Section 3 we review an efficient architecture for implementing these filters. In Section 4 we present the results of these filters for SAR target data from the Moving and Stationary Target Acquisition and Recognition (MSTAR) data set,[14] and in Section 5 we summarize the conclusions and present suggestions for further research.

2. Designing Quadratic Correlation Filters

Consider $\mathbf{x}$ and $\mathbf{y}$ to be two different random variables, each a d-dimensional vector (i.e., $\{\mathbf{x}, \mathbf{y}\} \in \Re^d$). Note that a $d_1 \times d_2$ image is ordered lexigraphically into a $d = d_1 d_2$ vector for the remainder of the analysis. An occurrence of random variable $\mathbf{x}$ will be considered a target of interest, and that of random variable $\mathbf{y}$ will be considered uninteresting clutter. Consider $\{x_i\}_{i=1}^{N_x}$ to be a sample from the target random variable $\mathbf{x}$ and $\{y_i\}_{i=1}^{N_Y}$ to be a sample from the clutter random variable $\mathbf{y}$. It is this *a priori* information (or training data) that we shall use to design the desired filters. Before we introduce a quadratic model with which to address the detection problem, a brief review of the linear model and its SDF solution is in order.

A. Review of Linear Synthetic Discriminant Functions

Linear SDFs were introduced in 1980 by Hester and Casasent.[12] That earlier form was designed as a linear combination of the training images and was referred to as a projection SDF. The projection SDF design algorithm is as follows: It is desired to use the training data to define a linear filter, $h \in \Re^d$, such that

$$\mathbf{x}^T\mathbf{h} = \begin{cases} \geq \alpha & x \text{ is a target} \\ < \alpha & x \text{ is not a target} \end{cases}. \tag{1}$$

To design such a filter we let $\mathbf{X}$ be the $d \times N_x$ matrix with the ith column made from the data x_i and use a similar definition for matrix $\mathbf{Y}$. In this context, the standard bracketing notation is used to indicate a matrix:

$$\begin{aligned} \mathbf{X} &= \lfloor \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{N_x} \rfloor, \\ \mathbf{Y} &= \lfloor \mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}_{N_y} \rfloor. \end{aligned} \tag{2}$$

With these matrices defined, the design criterion given by Eqs. (1) can be written as

$$[\mathbf{X} \quad \mathbf{Y}]^T\mathbf{h} = \mathbf{c} = \left[\underbrace{\mathbf{1} \quad \mathbf{1} \quad \cdots \quad \mathbf{1}}_{N_x} \underbrace{\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}}_{N_y} \right]^T, \tag{3}$$

and the value of α chosen to be somewhere between 0 and 1, depending on the performance goals of the specific application or system. Assuming that the desired filter $\mathbf{h}$ is in the linear span of the training data,

$$\mathbf{h} = [\mathbf{X} \quad \mathbf{Y}]\mathbf{a} \tag{4}$$

for some vector $\mathbf{a}$. Substituting Eq. (4) into Eq. (3) leads to a solution for $\mathbf{h}$:

$$\mathbf{h} = [\mathbf{X} \quad \mathbf{Y}]([\mathbf{X} \quad \mathbf{Y}]^T[\mathbf{X} \quad \mathbf{Y}])^{-1}\mathbf{c}. \tag{5}$$

If the correlation matrix of the data is a full-rank matrix, there is a unique solution for filter $\mathbf{h}$. If the data correlation matrix is less than full rank, then there are many solutions for $\mathbf{h}$. It is well known that using a reduced rank pseudoinverse calculation in Eq. (5) will lead to the unique solution for $\mathbf{h}$ that is

perpendicular to the null space of the data correlation matrix. If the dimension of $\mathbf{c}$ (number of training images) is greater than the dimension of the desired filter $\mathbf{h}$, the solution to Eq. (5) will yield an approximate solution of Eq. (3). Inasmuch as the data in matrix $\mathbf{Y}$ represent nontarget clutter (or noise), it makes sense to attempt to limit the variance of the filter in response to this clutter such that the probability that $\mathbf{Y}^T\mathbf{h} < 1$ is maximized. Thus we wish to minimize the variance of the filter output that is due to clutter,

$$E[(\mathbf{y}^T\mathbf{h})^2] = E[\mathbf{h}^T\mathbf{y}\mathbf{y}^T\mathbf{h}] = \mathbf{h}^T E[\mathbf{y}\mathbf{y}^T]\mathbf{h} \approx \mathbf{h}^T\mathbf{Y}\mathbf{Y}^T\mathbf{h}, \tag{6}$$

under the same constraints given by Eq. (3). This is the minimum-variance SDF solution[13] and leads to

$$\mathbf{h} = (\mathbf{Y}\mathbf{Y}^T)^{-1}[\mathbf{X} \quad \mathbf{Y}]([\mathbf{X} \quad \mathbf{Y}]^T(\mathbf{Y}\mathbf{Y}^T)^{-1}[\mathbf{X} \quad \mathbf{Y}])^{-1}\mathbf{c}. \tag{7}$$

The most notable shortcoming of the techniques described here is that they are linear operators. The set of all targets of interest in the detection problem is a large set. To assume that a single linear operator exists that is capable of approximating the desired behavior from Eqs. (1) is not realistic. In most applications, it is required to design a bank of linear filters, each of which is designed to detect a certain subset of the targets of interest. For example, to detect a particular vehicle, one might design a linear filter to detect that vehicle for a broadside view, a separate filter for the head-on view, a separate filter for the end-on view, etc. until all the angles are covered by filters. This leads to the design of many linear filters, each of which is designed completely independently from the others. One can account for the interdependencies of these filters by assuming up front that the bank of filters desired constitutes one quadratic filter.

B. Quadratic Synthetic Discriminant Function

Consider a quadratic design problem; it is desired to find a $d \times d$ matrix $\mathbf{T}$ such that

$$\mathbf{x}^T\mathbf{T}\mathbf{x} = \begin{cases} \geq\alpha & x \text{ is a target} \\ <\alpha & x \text{ is not a target} \end{cases}. \tag{8}$$

For a QSDF solution, we must choose reasonable constraint equations. Toward that goal, observe that

$$\begin{aligned} &[\mathbf{X} \quad \mathbf{Y}]^T\mathbf{T}[\mathbf{X} \quad \mathbf{Y}] = \mathbf{C}, \\ \Leftrightarrow\ &[\mathbf{X} \quad \mathbf{Y}]^T[\mathbf{T}\mathbf{X} \quad \mathbf{T}\mathbf{Y}] = \mathbf{C}, \\ \Leftrightarrow\ &\begin{bmatrix} \mathbf{X}^T\mathbf{T}\mathbf{X} & \mathbf{X}^T\mathbf{T}\mathbf{Y} \\ \mathbf{Y}^T\mathbf{T}\mathbf{X} & \mathbf{Y}^T\mathbf{T}\mathbf{Y} \end{bmatrix} = \mathbf{C}. \end{aligned} \tag{9}$$

We are interested only in the diagonal elements of matrix $\mathbf{C}$. All other entries can be arbitrary, as they are cross terms and will never actually be computed in practice. One possible choice for $\mathbf{C}$ could be

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix. This would force the response of the quadratic filter to be 1 when a target is present, -1 when clutter is present, and 0 for the cross terms. This equation may be overly restrictive, but it does have a solution:

$$\mathbf{T} = (\mathbf{X}\mathbf{X}^T + \mathbf{Y}\mathbf{Y}^T)^{-1}(\mathbf{X}\mathbf{X}^T - \mathbf{Y}\mathbf{Y}^T)(\mathbf{X}\mathbf{X}^T + \mathbf{Y}\mathbf{Y}^T)^{-1}. \tag{11}$$

If we let R_x and R_y be the correlation matrices for the target and the clutter training samples, respectively, then

$$\mathbf{T} = (\mathbf{R}_\mathbf{X} + \mathbf{R}_\mathbf{Y})^{-1}(\mathbf{R}_\mathbf{X} - \mathbf{R}_Y)(\mathbf{R}_\mathbf{X} + \mathbf{R}_\mathbf{Y})^{-1}. \tag{12}$$

This design criterion matches the original linear SDF design criterion in Eq. (5). We can extend the QSDF design to include a minimum-output-variance requirement as for the linear operators described by Eq. (7).

C. Minimum-Variance QSDF

Realizing that the linear minimum-variance SDF has been shown to be much more robust than the standard SDF design, we desire to incorporate the same constraints into the quadratic filter design. When the QCF $\mathbf{T}$ designed from Eq. (12) is implemented, the average output that is due to the target class will be approximately 1 and the average output that is due to the clutter class will be approximately -1. It is possible that the variance of the output of the filter will be so large that the performance of the filter (probability of error) will be unacceptable no matter what value of α is chosen to implement Eq. (8). Using the same design criterion as in the linear case (minimizing the output variance that is due to clutter) becomes difficult because the output variance of the quadratic filter is a fourth-order moment with respect to the data. Therefore a simplified constraint is used: Let $\mathbf{m}_x$ be the average of the target samples and $\mathbf{m}_y$ be the average of the clutter samples, and consider that we want to design a QSDF, $\mathbf{T}$, such that $E[\mathbf{x}^T\mathbf{T}\mathbf{x} - \mathbf{m}_x{}^T\mathbf{T}\mathbf{m}_x]$ and $E[y^T\mathbf{T}\mathbf{y} - \mathbf{m}_Y{}^T\mathbf{T}\mathbf{m}_Y]$ are both close to zero. These constraints will ensure that there is little variability among the classes. Thus a full SDF-like solution is possible. Let

$$\begin{aligned} \mathbf{X}_\mathbf{C} &= [\mathbf{x}_1 - \mathbf{m}_\mathbf{X} \quad \mathbf{x}_2 - \mathbf{m}_X \quad \cdots \quad \mathbf{x}_{N_x} - \mathbf{m}_\mathbf{X}], \\ \mathbf{Y}_\mathbf{C} &= [\mathbf{y}_1 - \mathbf{m}_\mathbf{Y} \quad \mathbf{y}_2 - \mathbf{m}_\mathbf{Y} \quad \cdots \quad \mathbf{y}_{\mathbf{N}_y} - \mathbf{m}_\mathbf{Y}], \end{aligned} \tag{13}$$

denote the centered data matrices. Then we wish the response to $\mathbf{X}_C$ and $\mathbf{Y}_C$ to be zero:

$$\begin{aligned} &[\mathbf{X} \quad \mathbf{X}_C \quad \mathbf{Y} \quad \mathbf{Y}_C]^T\mathbf{T}[\mathbf{X} \quad \mathbf{X}_C \quad \mathbf{Y} \quad \mathbf{Y}_C] = \mathbf{C}, \\ \Leftrightarrow\ &[\mathbf{X} \quad \mathbf{X}_C \quad \mathbf{Y} \quad \mathbf{Y}_C]^T[\mathbf{T}\mathbf{X} \quad \mathbf{T}\mathbf{X}_C \quad \mathbf{T}\mathbf{Y} \quad \mathbf{T}\mathbf{Y}_C] = \mathbf{C}, \\ \Leftrightarrow\ &\begin{bmatrix} \mathbf{X}^T\mathbf{T}\mathbf{X} & \mathbf{X}^T\mathbf{T}\mathbf{X}_C & \mathbf{X}^T\mathbf{T}\mathbf{Y} & \mathbf{X}^T\mathbf{T}\mathbf{Y}_C \\ \mathbf{X}_C{}^T\mathbf{T}\mathbf{X} & \mathbf{X}_C{}^T\mathbf{T}\mathbf{X}_C & \mathbf{X}_C{}^T\mathbf{T}\mathbf{Y} & \mathbf{X}_C{}^T\mathbf{T}\mathbf{Y}_C \\ \mathbf{Y}^T\mathbf{T}\mathbf{X} & \mathbf{Y}^T\mathbf{T}\mathbf{X}_C & \mathbf{Y}^T\mathbf{T}\mathbf{Y} & \mathbf{Y}^T\mathbf{T}\mathbf{Y}_C \\ \mathbf{Y}_C{}^T\mathbf{T}\mathbf{X} & \mathbf{Y}_C{}^T\mathbf{T}\mathbf{X}_C & \mathbf{Y}_C{}^T\mathbf{T}\mathbf{Y} & \mathbf{Y}_C{}^T\mathbf{T}\mathbf{Y}_C \end{bmatrix} = \mathbf{C}. \end{aligned} \tag{14}$$

Again, we are interested only in the terms on the main diagonal. The off-diagonal terms are never encountered when one is implementing Eq. (8) and thus it would be ideal to ignore the off-diagonal terms when one is designing a solution. For our design we chose to constrain the off-diagonal element to be zero; thus we define the constraint matrix as

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{15}$$

Solving for $\mathbf{T}$ with left and right pseudoinverses yields

$$\mathbf{T} = (\mathbf{R}_X + \mathbf{C}_X + \mathbf{R}_Y + \mathbf{C}_Y)^{-1}(\mathbf{R}_X - \mathbf{R}_Y) \times (\mathbf{R}_X + \mathbf{C}_X + \mathbf{R}_Y + \mathbf{C}_Y)^{-1}, \tag{16}$$

where $\mathbf{C}_x$ and $\mathbf{C}_y$ are the covariance matrices for the target and the clutter samples, respectively. These QSDF solutions to the QCF design problem are approximations, in the least-squares sense, to Eqs. (9) and (14).

D. Subspace QSDF

It is clear that there is approximation power being lost in these designs by approximating all the off-diagonal terms that appear in the constraint matrix and would never appear in an implemented filter. Toward the goal of designing a QSDF solution without this drawback, we require an additional assumption for matrix $\mathbf{T}$. Akin to the situation for linear filters, for which it is assumed that filter $\mathbf{h}$ is in the linear span of the data vectors, we assume that matrix $\mathbf{T}$ is in the linear span of the outer product of the data vectors.

Again, consider $\{x_i\}_{i=1}^{N_x}$ to be a sample from the target class and $\{y_i\}_{i=1}^{N_Y}$ to be a sample from the clutter class. Also, let $\{z_i\}_{i=1}^{N}$ be the union of these two sets of data, where $N = N_x + N_y$. Let $\mathbf{X}$ be the $d \times N_X$ matrix with the ith column made up from the data x_i with similar definitions for the matrices $\mathbf{Y}$ and $\mathbf{Z}$:

$$\begin{aligned} \mathbf{X} &= \lfloor \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{N_x} \rfloor, \\ \mathbf{Y} &= [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}N_y], \\ \mathbf{Z} &= [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N]. \end{aligned} \tag{17}$$

It is desired to find a matrix $\mathbf{T}$ that satisfies Eq. (8). Let $\mathbf{T}$ be a linear combination of the outer products of the data samples:

$$\mathbf{T} = \sum_{i=1}^{N} \beta_i z_i z_i^T \tag{18}$$

for some coefficient sequence $\{\beta_i\}_{i=1}^{N}$. The problem then becomes solving for the vector $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_N]$ such that operator $\mathbf{T}$ satisfies certain constraints. We wish to constrain the expected value of the output of the quadratic filter to be 1 when a target is present and to be -1 when clutter is present. Defining the conditional expectation operator $E_X[\ ]$ to be the expected value when a target is present and defining $E_Y[\ ]$ similarly, we can represent the desired constraints as

$$\begin{aligned} E_X[\lfloor \mathbf{z}^T\mathbf{T}\mathbf{z} \rfloor] &= 1, \\ E_Y[\mathbf{z}^T\mathbf{T}\mathbf{z}] &= -1. \end{aligned} \tag{19}$$

Now that the expected value or mean has been constrained, we wish to control the variance of the output to be as small as possible. To design constraints on the output variance, we again want to avoid fourth-order moments on the data; thus we constrain all the training data to be as close to the mean value as possible. This leads to $N = N_X + N_Y$ constraints

$$\begin{aligned} x_i^T\mathbf{T}x_i - E_X[\mathbf{z}^T\mathbf{T}\mathbf{z}] &= 0 \qquad \forall\, i = 1, \ldots, N_X, \\ y_i^T\mathbf{T}y_i - E_Y[\mathbf{z}^T\mathbf{T}\mathbf{z}] &= 0 \qquad \forall\, i = 1, \ldots, N_Y. \end{aligned} \tag{20}$$

If they are satisfied by $\mathbf{T}$, the constraints given in Eqs. (19) and (20) will yield perfect performance from Eq. (8) for a threshold of $\alpha = 0$ when the test data are well represented by the training. Of course, in practice, these constraints must be approximated. Beginning with the constraints on the output mean value, we substitute the model for $\mathbf{T}$ from Eq. (18) into the constraints from Eqs. (19). This leads to

$$\begin{aligned} E_X[\mathbf{z}^T\mathbf{T}\mathbf{z}] &= E_X\left(z^T \sum_{i=1}^{N} \beta_i z_i z_i^T z\right) = E_X\left(\sum_{i=1}^{N} \beta_i z^T z_i z_i^T z\right) \\ &= \sum_{i=1}^{N} \beta_i z_i^T E_X(zz^T) z_i \approx \sum_{i=1}^{N} \beta_i z_i^T \mathbf{R}_\mathbf{x} z_i = 1, \\ E_Y[\mathbf{z}^T\mathbf{T}\mathbf{z}] &= E_Y\left(z^T \sum_{i=1}^{N} \beta_i z_i z_i^T z\right) = E_Y\left(\sum_{i=1}^{N} \beta_i z^T z_i z_i^T z\right) \\ &= \sum_{i=1}^{N} \beta_i z_i^T E_Y(zz^T) z_i \approx \sum_{i=1}^{N} \beta_i z_i^T \mathbf{R}_\mathbf{Y} z_i = -1, \end{aligned} \tag{21}$$

where $\mathbf{R}_X$ and $\mathbf{R}_Y$ are the correlation matrices of the target and the clutter classes, respectively. We can also expand the variability constraints given in expressions (21) by substituting Eq. (18) into the constraint equations to yield

$$\begin{aligned} \mathbf{x_i}^T\mathbf{T}\mathbf{x}_i - E_X(\mathbf{z}^T\mathbf{T}\mathbf{z}) &\approx \sum_{j=1}^{N} \beta_j z_j^T (x_i x_i^T - \mathbf{R}_\mathbf{X}) z_j \\ &= 0 \qquad \forall\, i = 1, \ldots, N_X, \\ \mathbf{y_i}^T\mathbf{T}\mathbf{y}_i - E_Y(\mathbf{z}^T\mathbf{T}\mathbf{z}) &\approx \sum_{j=1}^{N} \beta_j z_j^T (y_i y_i^T - \mathbf{R}_\mathbf{Y}) z_j \\ &= 0 \qquad \forall\, i = 1, \ldots, N_Y. \end{aligned} \tag{22}$$

Expressions (21) and (22) lead to $N + 2$ constraint equations that must be solved for the N coefficients in the model given by Eq. (18). Because this is an underdetermined system, an exact solution may not be possible, and we must resort to an approximate so-

lution. We can obtain a least-squares solution by formulating the auxiliary vectors

$$\begin{aligned}
\mathbf{p} &= [\mathbf{z}_1{}^T\mathbf{R}_X\mathbf{z}_1 \quad \mathbf{z}_2{}^T\mathbf{R}_X\mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N{}^T\mathbf{R}_X\mathbf{z}_N]^T,\\
\mathbf{q} &= [\mathbf{z}_1{}^T\mathbf{R}_Y\mathbf{z}_1 \quad \mathbf{z}_2{}^T\mathbf{R}_Y\mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N{}^T\mathbf{R}_Y\mathbf{z}_N]^T,\\
\mathbf{p_i} &= [\mathbf{z}_1{}^T(\mathbf{x}_i\mathbf{x}_i{}^T - \mathbf{R_X})\mathbf{z}_1 \quad \mathbf{z}_2{}^T(\mathbf{x}_i\mathbf{x}_i{}^T\\
&\quad - \mathbf{R}_X)\mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N{}^T(\mathbf{x}_i\mathbf{x}_i{}^T - \mathbf{R}_X)\mathbf{z}_N]^T \quad \forall\mathbf{i},\\
\mathbf{q_i} &= [\mathbf{z}_1{}^T(\mathbf{y}_i\mathbf{y}_i{}^T - \mathbf{R}_Y)\mathbf{z}_1 \quad \mathbf{z}_2{}^T(\mathbf{y}_i\mathbf{y}_i{}^T\\
&\quad - \mathbf{R}_Y)\mathbf{z}_2 \quad \cdots \quad \mathbf{z}_N{}^T(\mathbf{y}_i\mathbf{y}_i{}^T - \mathbf{R}_Y)\mathbf{z}_N]^T \quad \forall\mathbf{i},\\
\mathbf{c} &= [\mathbf{1} \quad -\mathbf{1} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]^T,\\
\beta &= [\beta_1 \quad \cdots \quad \beta_N]^T. \qquad (23)
\end{aligned}$$

With these auxiliary vectors defined, the constraints given in expressions (21) and (22) can be written in compact form as

$$\mathbf{B}\boldsymbol{\beta} = \mathbf{c}, \qquad (24)$$

where

$$\mathbf{B} = [\mathbf{p} \quad \mathbf{q} \quad \mathbf{p}_1 \quad \cdots \quad \mathbf{p}_{N_X} \quad \mathbf{q}_1 \quad \cdots \quad \mathbf{q}_{N_Y}]^T. \qquad (25)$$

Thus, the minimum norm least-squares solution for the unknown coefficient vector is

$$\boldsymbol{\beta} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{c}. \qquad (26)$$

This gives a solution to Eq. (18) that results in a QSDF filter as

$$\mathbf{T} = \mathbf{Z}\Delta\mathbf{Z}^T. \qquad (27)$$

Here Δ is an $N \times N$ diagonal matrix with $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_N]$ [from Eq. (26)] along the main diagonal. Equation (27) leads to a subspace QSDF filter similar in design to the linear minimum-variance SDF filter from Eq. (7). As mentioned above, a quadratic model for building correlation filters for target detection has the advantage over linear filter design techniques in that all the channels of the quadratic filter are designed simultaneously rather than independently as in the linear case. However, the linear filters have an obvious algorithm for implementing the filter over all pixels in an entire image through correlations. The ease of implementation of the quadratic filtering model for an entire image is not so obvious and is presented in Section 3.

3. Implementation of a QSDF

To implement a quadratic filter as represented by Eq. (8) requires an efficient computational representation of the filter to make the computational load reasonable for standard image-processing applications. A methodology for this implementation was presented by Mahalanobis *et al.*[11] that assumes that **T** is symmetric and less than full rank. The solution for **T** given by Eq. (27) is clearly symmetric. However, there is nothing in the design that controls the rank of the matrix, which is key to this efficient implementation. Later in this section, a reduced-rank approximation to **T** will be presented that maps into the general implementation methodology. As a review of the implementation methodology presented by Mahalanobis *et al.*,[11] we first assume that **T** is a symmetric and nonnegative matrix of rank N. Under this assumption, **T** can be expressed as $\mathbf{T} = \mathbf{F}\mathbf{F}^T$, where $\mathbf{F} = [\mathbf{f}_1 \quad \mathbf{f}_2 \quad \cdots \quad \mathbf{f}_N]$ is a $d \times N$ matrix ($N \leq d$). We note that the output of the quadratic filter can be expressed as

$$\begin{aligned}
y &= \mathbf{z}^T\mathbf{T}\mathbf{z} = \mathbf{z}^T\mathbf{F}\mathbf{F}^T\mathbf{z}\\
&= \mathbf{v}^T\mathbf{v}. \qquad (28)
\end{aligned}$$

Here, $\mathbf{v} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_N]^T$ is a vector of the projection of **z** onto the N columns of **F**. Clearly, **v** is a function of the spatial region of the image that is represented by **z**. The question is how the elements of **v** can be computed over the entire scene by use of efficient signal-processing methods. Let $I(m, n)$ represent the full image to be processed. Furthermore, let us reorder the elements of $\mathbf{f}_i$ into a $d_1 \times d_2$ kernel $f_i(m, n)$. It is then easy to see that the values of v_i (i.e., the ith element of **v**) for all locations within the image can be obtained by means of the two-dimensional correlation of $f_i(m, n)$ and $x(m, n)$ as

$$v_i(m, n) = x(m, n) \otimes f_i(m, n), \qquad 1 \leq i \leq N, \quad (29)$$

where the symbol $\otimes$ indicates the two-dimensional correlation operation. Here, v_i is expressed with the indices (m, n) to indicate that it is a function of the location of **z** in the image.

We can view the samples of $v_i(m, n)$ as a partial result at each point in the image obtained by projecting the $d_1 \times d_2$ region of the image centered at that point on $\mathbf{f}_i$. To obtain the quadratic term in Eq. (28) for all points in the image, it now remains to square the pixels of the surfaces of partial results and add them. Therefore the output of the quadratic filter for all points in the image is given by

$$y(m, n) = \sum_{i=1}^{N} |v_i(m, n)|^2 = \sum_{i=1}^{N} |x(m, n) \otimes f_i(m, n)|^2. \qquad (30)$$

The expression in Eq. (30) is a succinct way to express the output of the quadratic filter in response to the full input image. The computations can be readily dealt with by use of N two-dimensional cross-correlation operations, each of which is efficiently implemented with fast Fourier transforms. Removing the assumption that **T** is nonnegative leads to a similar implementation because **T** can be represented by the difference of two nonnegative operators:

$$\mathbf{T} = \mathbf{F}\mathbf{F}^T - \mathbf{G}\mathbf{G}^T. \qquad (31)$$

Other than the fact that **T** here is a difference of two terms, the solution follows the same form as in the general case described in Eq. (30). The output of the

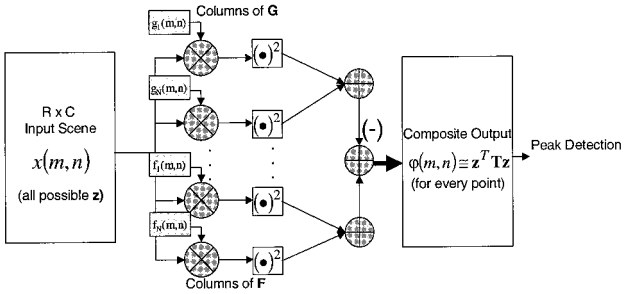

Fig. 1. Efficient architecture for implementing the quadratic filter correlates the input image with a bank of linear filters to yield partial results that are squared and added to produce the desired quadratic output for every point in the input image.

quadratic filter in response to the full image also involves two terms and takes the form

$$y(m, n) = \sum_{i=1}^{N_2} |x(m, n) \otimes f_i(m, n)|^2 - \sum_{i=1}^{N_1} |x(m, n) \otimes g_i(m, n)|^2, \quad (32)$$

where $f_i(m, n)$ and $g_i(m, n)$ are the ith columns of $\mathbf{F}$ and $\mathbf{G}$, respectively, each reshaped into a $d_1 \times d_2$ kernel.

Figure 1 depicts the architecture required for efficient processing of a full image with the quadratic filter. Essentially, the input image is correlated with a bank of linear filters to produce partial results that are squared and added to yield the desired quadratic output for every point in the input image. Although several branches of linear filters are required, the architecture shown comprises a single quadratic filter. Because each of the branches can be implemented by use of a fast Fourier transform, the number of multiplications required in the architecture in Fig. 1 is proportional to $N \log(d_1 d_2)$, where N is the number of branches and $d_1 \times d_2$ is the size of the filter kernels. In comparison, the number of multiplications required for the direct computation of the quadratic term at every point is proportional to $(d_1 d_2)^2$.

For the operators described by Eqs. (16) and (18), one must decompose $\mathbf{T}$ into the form of equation (31). Once the $\mathbf{F}$ and $\mathbf{G}$ matrices have been defined, the efficient computational architecture in Fig. 1 can be used to compute the quadratic filter over the entire image in an efficient manner. Because operators $\mathbf{T}$ from both Eqs. (16) and (27) are symmetric, the eigendecomposition of $\mathbf{T}$ will yield real eigenvalues:

$$\mathbf{T} = \mathbf{U}\Lambda\mathbf{U}^T. \quad (33)$$

Equation (33) shows the eigendecomposition of T, where the columns of $\mathbf{U}$ are the eigenvectors and matrix Λ is a diagonal matrix containing the eigenvalues of $\mathbf{T}$. If the eigenvalues that are close to zero are removed from Λ, and Λ is separated into the difference of two nonnegative matrices, then $\mathbf{T}$ can be decomposed in the following manner:

$$\mathbf{T} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m] \begin{bmatrix} \boldsymbol{\lambda}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \boldsymbol{\lambda}_m \end{bmatrix} [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m]^T - [\mathbf{u}_{d-n} \quad \mathbf{u}_{d-n+1} \quad \cdots \quad \mathbf{u}_d] \begin{bmatrix} -\boldsymbol{\lambda}_{d-n} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\boldsymbol{\lambda}_d \end{bmatrix} [\mathbf{u}_{d-n} \quad \mathbf{u}_{d-n+1} \quad \cdots \quad \mathbf{u}_d]^T. \quad (34)$$

Here λ_i is positive for $i = 1, \ldots, m$ and negative for $i = d - n, \ldots, d$. This fits the form of Eq. (31), where the columns of $\mathbf{F}$ are defined as $F_i = \sqrt{\lambda_i}\, u_i$ and the columns of $\mathbf{G}$ are $\mathbf{G}_i = \sqrt{-\lambda_{d-n+i-1}}\, u_{d-n+i-1}$. Therefore $\mathbf{T}$ can be approximated by a decomposition of the form in Eq. (31) and, thus, be implemented efficiently with the architecture described by Fig. 1 and Eq. (32). Several quadratic filters have been coded, trained, and tested on the public-domain MSTAR SAR dataset, and the results are presented in Section 4 below.

4. Example of Target Detection in SAR Imagery

The public-domain MSTAR data set[14] is a collection of SAR image data of various targets as well as a collection of SAR images of clutter scenes. The target chips are 256×256 images containing targets at various aspect and depression angles. A set of tank target chips (more precisely, a T72 tank) was extracted from these data for training purposes and cropped to 40×40 image chips. A standard bank of linear maximum average correlation height (MACH) filters[8] was trained on these targets and applied to the MSTAR clutter imagery. Local peaks in the MACH filters were extracted and cropped to 40×40 image chips to represent the clutter training samples. The reason for generating the clutter class in this manner is to test the algorithm's performance on dif-

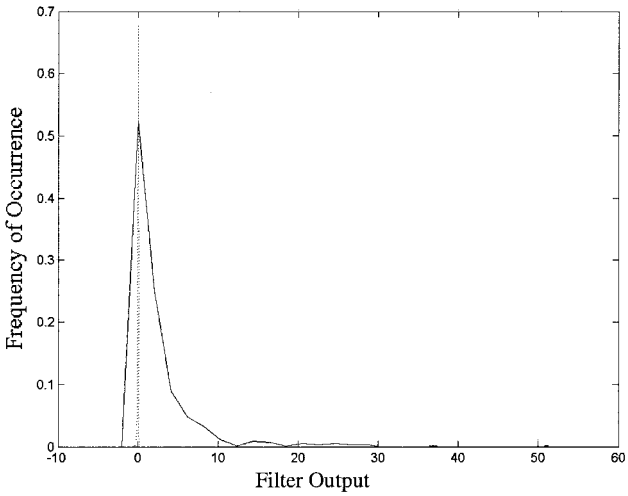

Fig. 2. FKQCF output: solid curve, target class; dotted curve, clutter class.

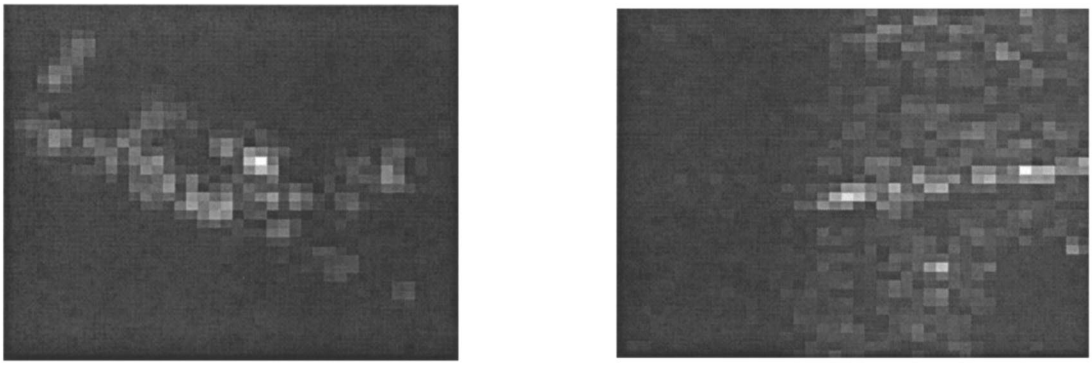

Fig. 3. Left, example of a target image used for Figs. 4, 7, 10, and 14 below. Right, example of a clutter image used for Figs. 4, 7, 10, and 14.

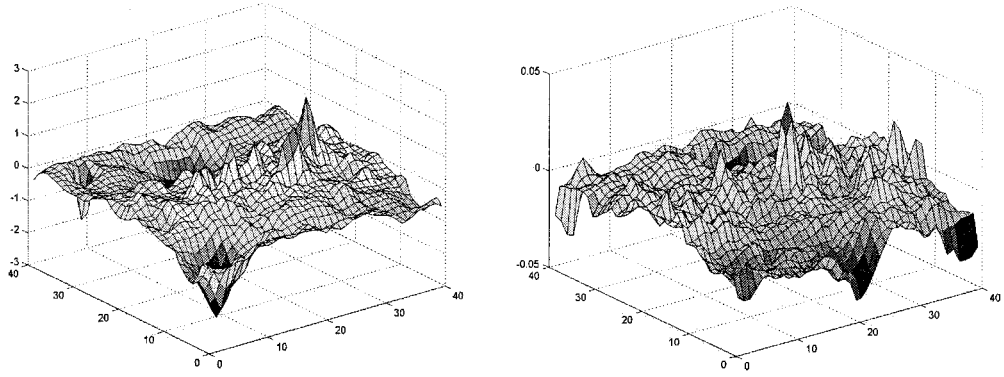

Fig. 4. Left, example FKQCF target correlation surface. Right, example FKQCF clutter correlation surface.

ficult clutter examples, not just random imagery, which could be benign and unchallenging. Also, using the MACH filter to generate the clutter class yields a qualitative indication of the increase in performance of the quadratic methods because we are testing directly on the false alarms of a powerful linear method.

With these training data, four quadratic target detection filters were designed: the QSDF filter from Eq. (16), the subspace QSDF (SSQSDF) filter from Eq. (25), the Fukanaga–Koontz QCF (FKQCF) derived in Ref. 10, and the Rayleigh quotient QCF (RQQCF) derived in Ref. 11. We then tested these filters on a completely different set of T72 image chips and clutter chips to get performance results.

A. Results of the FKQCF

The output of the FKQCF on the testing data is presented in Fig. 2 and shows that the class separation

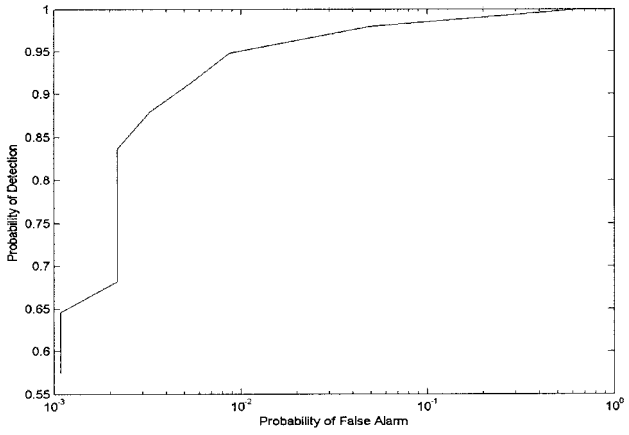

Fig. 5. ROC curve for a FKQCF for the MSTAR data test set.

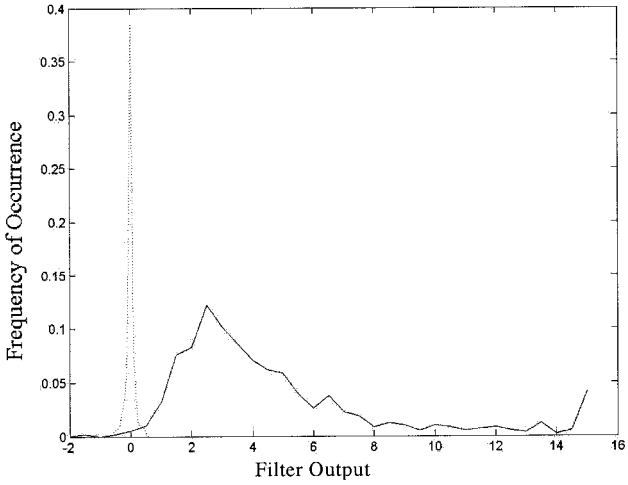

Fig. 6. Output responses of test data for a RQQCF. Solid curve, target; dashed curve, clutter.

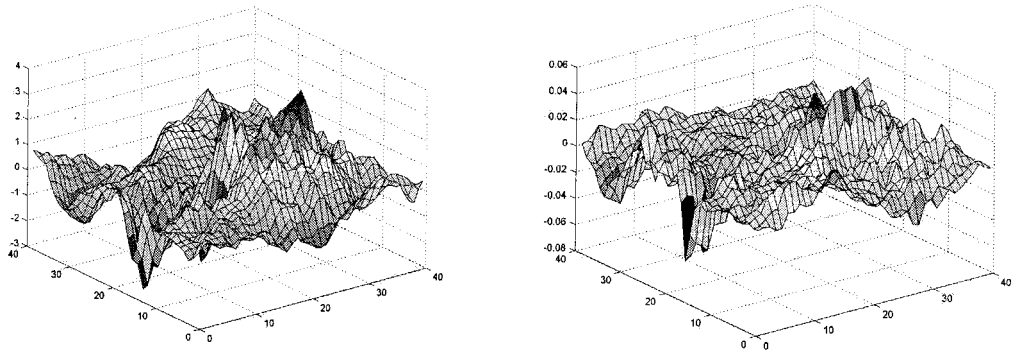

Fig. 7. Left, example RQQCF target correlation surface. Right, example RQQCF clutter correlation surface.

results are limited. In this context, the filter output is defined as the central correlation value.

Along with the ability to separate the classes, a relevant performance criterion for a correlation filter is the shape of the main peak. A sharp peak with suppressed sidelobes is desirable for robustness in target location accuracy. For the purposes of illustration, typical target and clutter images are presented in Fig. 3. Subsequent algorithm responses to these images are given below. The correlation surface responses of the FKQCF for these typical target and clutter examples are given in Fig. 4. They show that the correlation peak for the target is much higher than for the clutter. However, the target response exhibits large sidelobes, which will induce false alarms or a poor target location estimate. The most important performance statistics of a detection algorithm are the probability of detection and the probability of a false alarm. Typically, these statistics are presented on a receiver operating characteristics (ROC) curve, where the probability of detection is plotted as a function of the probability of a false alarm. The ROC curve for the FKQCF is presented in Fig. 5. This curve shows that, for a probability of false alarm of 0.01, one can expect to detect approximately 95% of the targets for these data.

B. Results of RQQCF

The RQQCF design exhibits better performance than the FKQCF as it is intended to achieve optimal class separation as opposed to optimal information content. For a treatment of the theory, see Mahalanobis *et al.*[11] The output response to the test data for this algorithm is presented in Fig. 6. There is clearly separation between the classes. Also, the correlation surface behavior is shown in Fig. 7.

As with the FKQCF, there is a significantly larger peak in the target response surface than in the clutter response surface. Also, there are still significant sidelobes in the target response surface, which might result in false alarms, poor target location estimation, or both. The final performance characterization, the ROC curve in Fig. 8, shows excellent performance, with a probability of detection of 0.99 obtained at a false-alarm rate of 0.01.

C. Results of QSDF

The QSDF design algorithm from Eq. (16) was designed with the training data and tested on the test data. Both $\mathbf{T}$ [from Eq. (16)] and $\hat{\mathbf{T}}$ [from Eq. (34)] were implemented. The responses of these filters to the testing data are shown in Fig. 9. These results show a significant loss of separation when $\mathbf{T}$ is approximated by $\hat{\mathbf{T}}$. Also, these results show that the hard constraints (target response = 1, clutter response = −1) are not approximated well in the final solution. The correlation surface responses show that the QSDF solution does not exhibit good performance (Fig. 10). It is apparent that there is a performance price to pay in this design by requiring so many hard constraints, most of which [e.g., the off-diagonal elements of $\mathbf{C}$ in Eqs. (14)] are never implemented in practice. The ROC curve for this algorithm is presented in Fig. 11 and shows that a probability of detection of 0.85 is attainable for a false-alarm rate of 0.01.

D. Results of SSQSDF

For this data set the SSQSDF had the best overall performance. Before characterizing this performance we must note that the matrix inversion of $\mathbf{B}^T\mathbf{B}$

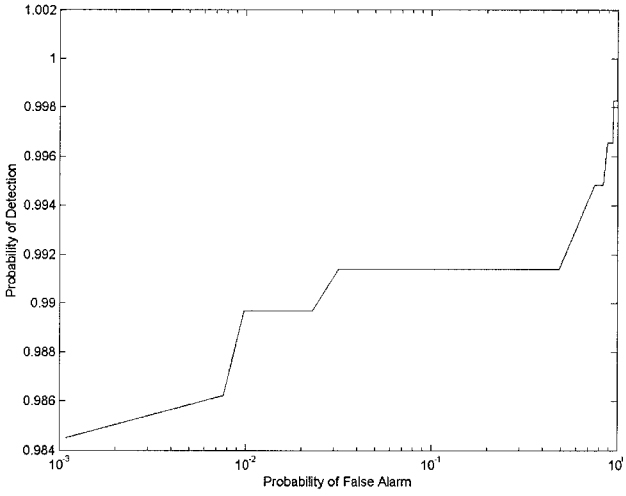

Fig. 8. ROC curve for a RQQCF.

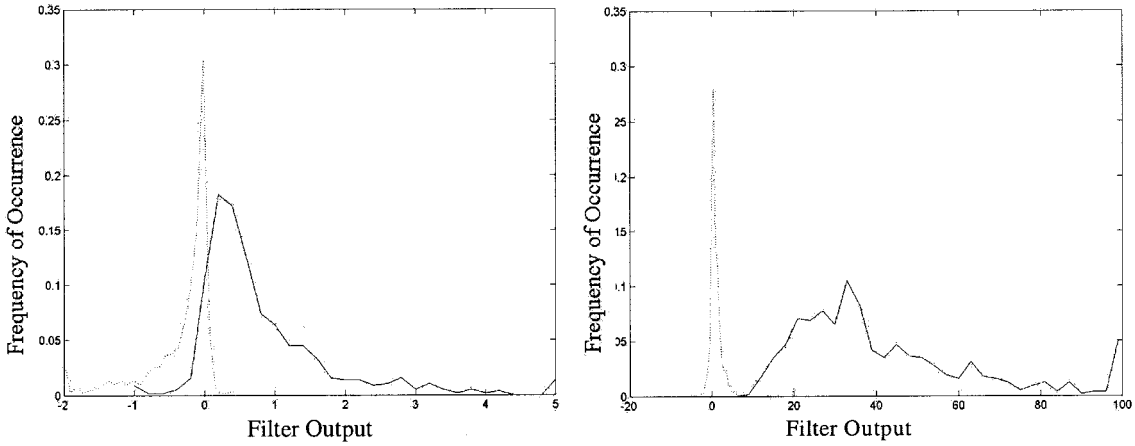

Fig. 9. Left, $\hat{\mathbf{T}}$ (QSDF) response to test data; right, $\mathbf{T}$ (full-rank QSDF) response. Solid curves, targets; dashed curves, clutter.

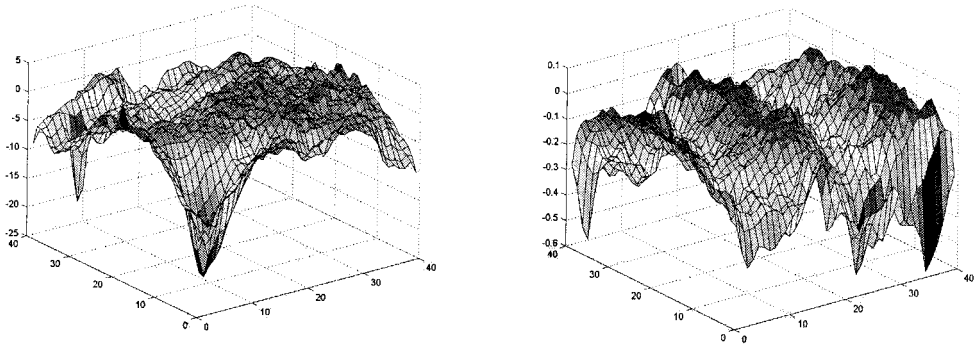

Fig. 10. Left, example QSDF target correlation surface; right, example QSDF clutter correlation surface.

from Eq. (26) turns out to be ill conditioned. A log plot of the singular values of $\mathbf{B}^T\mathbf{B}$ is presented in Fig. 12, along with the standard threshold level for a reduced-rank inversion algorithm that is typical of the Matlab programming language.

Because a reduced-rank inverse is being applied, the constraints of Eq. (24) will not necessarily be satisfied exactly for the training data. Histograms of the output responses for $\mathbf{T}$ [Eq. (27)] and $\hat{\mathbf{T}}$ [Eq. (34)] are given in Fig. 13 for the training and testing data sets. The figure at bottom left shows that the full-rank SSQSDF quadratic filter satisfies most of the constraints given in Eq. (24) precisely. The only constraint not satisfied is the average clutter response $\mathbf{q}^T\boldsymbol{\beta} = -1$. The figure at the bottom right shows that the design criteria generalize well from the training to the test data sets; the top two figures show that the approximation used for efficient implementation [Eq. (34)] does not much affect the separation of the classes. The correlation surfaces also show excellent performance and are shown in Fig. 14. There is a clear peak in the target response correla-

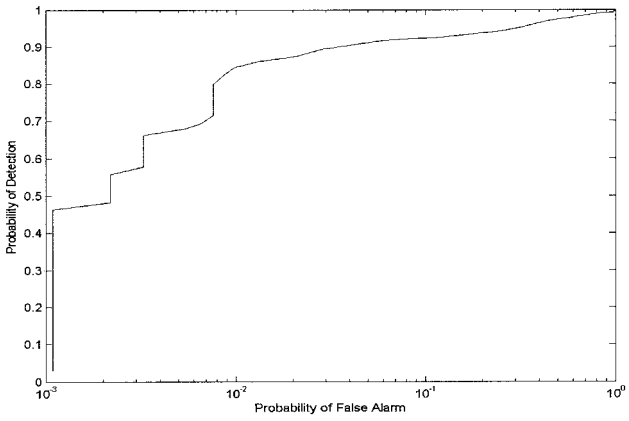

Fig. 11. ROC curve for a QSDF algorithm.

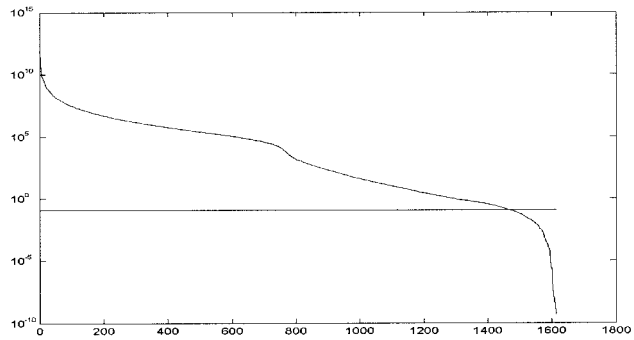

Fig. 12. Singular values of $\mathbf{B}^T\mathbf{B}$. Horizontal line, threshold value for reduced-rank inverse.

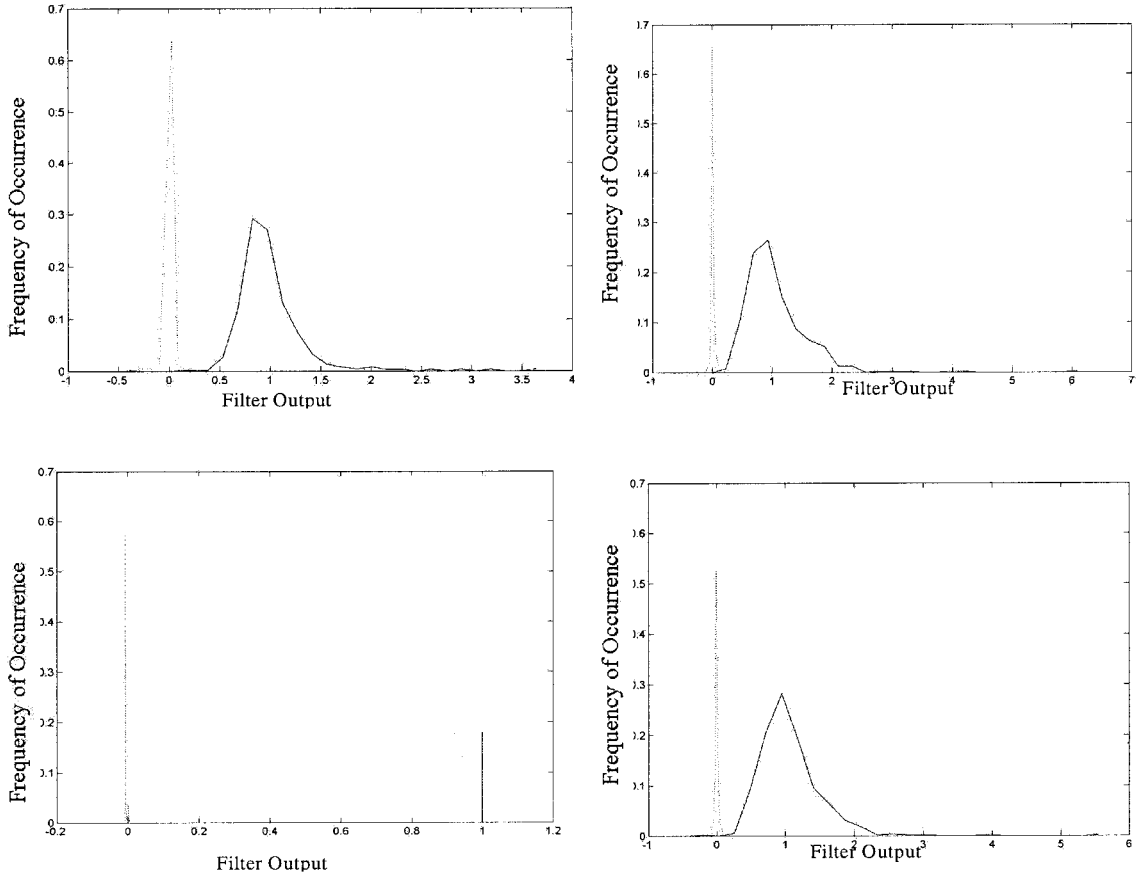

Fig. 13. Top left, clutter and target output responses from a SSQSDF for the training data. Top right, clutter and target output responses from a SSQSDF for the test data. Bottom left, clutter and target output responses from the full-rank SSQSDF for the training data. Bottom left, clutter and target output responses from a full-rank SSQSDF for the test data.

tion surface, with suppressed sidelobes and small-magnitude energy in the correlation surface response to the clutter example. This should lead to good probability of detection and probability of false-alarm performance statistics.

Figure 15 presents the ROC curve results and shows nearly perfect performance, with a probability of detection of 1.0 possible at a false-alarm rate of 0.002. These results are clearly better than for all the other techniques on this data set. This test shows that the subspace QSDF from Eq. (27) has perfect performance and that the RQQCF performs nearly as well. The FKQCF and the QSDF of Eq. (16) exhibit the poorest performance. These results are based on both **F** and **G** containing four basis functions each. Therefore the computations are identical for each filter. It should be noted here that the reduced performance of the FKQCF and the

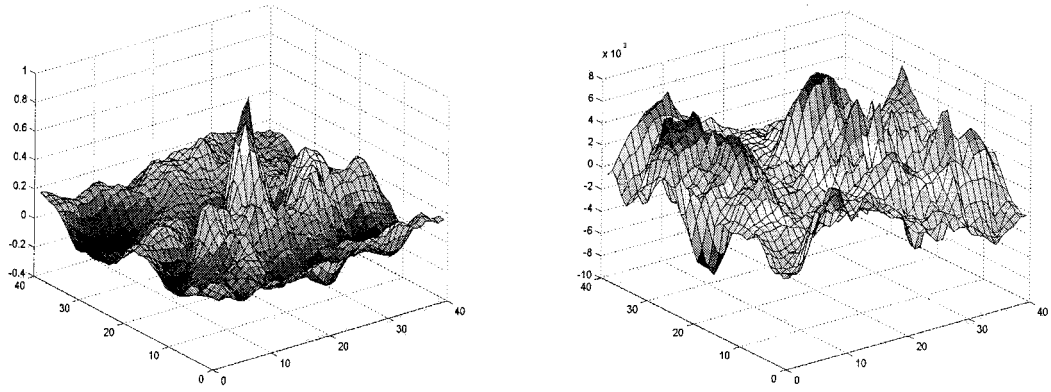

Fig. 14. Left, example SSQSDF target correlation surface; right, example SSQSDF clutter correlation surface.

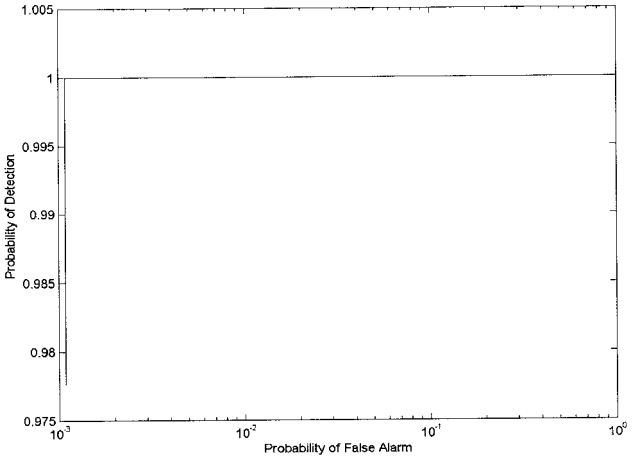

Fig. 15. ROC curve for a SSQSDF design.

QSDF can be traced to the rank deficiencies in the matrix inversions for each design. A trial-and-error, hands-on approach to reduced-rank inversions can and does improve the performance, but no systematic approach to choosing the optimal rank reduction is available. Experimentation has shown that the RQQCF does not exhibit as much performance variability related to performing a reduced-rank matrix inversion, and the subspace QSDF does not seem to be sensitive to the rank of the B matrix. Although the SSQSDF seems to have this desirable property and also exhibited the best performance in this test, the matrices involved have dimensions equal to the number of training chips, rather than to the dimensions of the training chips as the other techniques do. This can lead to prohibitively large matrices for the SSQSDF and the requirement of an adaptive training algorithm (the subject of further research) for large datasets.

5. Conclusions

Two methodologies for designing quadratic correlation filters for target detection in image data have been developed that extend the synthetic discriminant function algorithms for linear filters. Both derivations fit efficient computational architecture presented in previous literature. The algorithms were trained and tested on MSTAR public-domain imagery with probabilities of detection and probabilities of false alarms compared with two other recently derived QCFs for target detection. The results are interesting and show that the SSQSDF is the design technique with the best performance in this test.

To train the SSQSDFs, large matrices have been built and need to be inverted. A systematic adaptive training algorithm is a valuable topic for further research, as is a systematic methodology for a rank-reduced inversion for the QSDF algorithm.

This research was sponsored by the U.S. Army Research Office. Financial support was received from the Defense Advanced Research Projects Agency under contract DAAD19-02-C-0025. The contents of this paper do not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

References

1. J. Starch, R. Sharma, and S. Shaw, "A unified approach to feature extraction for model based ATR," in *Algorithms for Synthetic Aperture Radar Imagery III*, E. G. Zelnio and R. J. Douglass, eds., Proc. SPIE **2757,** 294–305 (1996).
2. D. Casasent and R. Shenoy, "Feature space trajectory for distorted object classification and pose estimation in SAR," Opt. Eng. **36,** 2719–2728 (1997).
3. B. Bhanu and J. Ahn, "A system for model-based recognition of articulated objects," in *International Conference on Pattern Recognition* (IEEE Computer Society, Washington, D.C., 1998), pp. 1812–1815, 1998).
4. S. Z. Der, Q. Zheng, R. Chellappa, B. Redman, and Hesham 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, New York, 2002), pp. 151–187.
5. L. A. Chan, S. Z. Der, and N. M. Nasrabadi, "Neural based target detectors for multi-band infrared imagery," in *Image Recognition and Classification, Algorithms, Systems and Applications*, B. Javidi, ed. (Marcel Dekker, New York, 2002), pp. 1–36.
6. D. Torreiri, "A linear transform that simplifies and improves neural network classifiers," presented at the International Conference on Neural Networks, Washington, D.C., 1996.
7. V. Page, F. Goudail, and P. Réfrégier, "Improved robustness of target location in nonhomogeneous backgrounds by use of the maximum-likelihood ratio test location algorithm," Opt. Lett. **24,** 1383–1385 (1999).
8. B. V. K. Vijaya Kumar, "Tutorial survey of composite filter designs for optical correlators," Appl. Opt. **31,** 4773–4801 (1992).
9. A. Mahalanobis, B. V. K. Vijaya Kumar, S. R. F. Sims, and J. Epperson, "Unconstrained correlation filters," Appl. Opt. **33,** 3751–3759 (1994).
10. A. VanderLugt, "Signal detection by complex spatial filtering," IEEE Trans. Inf. Theory **10,** 139–145 (1964).
11. A. Mahalanobis, R. Muise, S. R. Stanfill, and A. Van Nevel, "Design and application of quadratic correlation filters for target detection," IEEE Trans. Aerosp. Eng. (to be published).
12. C. F. Hester and D. Casasent, "Multivariant technique for multiclass pattern recognition," Appl. Opt. **19,** 1758–1761 (1980).
13. B. V. K. Vijaya Kumar, "Minimum variance synthetic discriminant functions," J. Opt. Soc. Am. A **3,** 1579–1584 (1986).
14. Three Class Public MSTAR data set, CD version, released for the Defense Advanced Research Projects Agency by Veda, Inc., the Dayton Group, 1997.
15. A. Mahalanobis, L. Ortiz, and B. V. K. Vijaya Kumar, "Performance of the MACH/DCCF algorithms on the 10-class public release MSTAR data set," in *Algorithms for Synthetic Aperture Radar Imagery VI*, E. G. Zelnio, eds., Proc. SPIE **3721,** 285–291 (1999).