

Selecting a composite correlation filter design: a survey and comparative study

Ryan A. Kerekes

Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
E-mail: kerekesra@ornl.gov

B. V. K. Vijaya Kumar

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract. Many composite correlation filter designs have been proposed for solving a wide variety of target detection and pattern recognition problems. Due to the large number of available designs, however, it is often unclear how to select the best design for a particular application. We present a theoretical survey and an empirical comparison of several popular composite correlation filter designs. Using a database of rotational target imagery, we show that some such filter designs appear to be better choices than others under computational and performance constraints. We compare filter performance in terms of noise tolerance, computational load, generalization ability, and distortion in order to provide a multifaceted examination of the characteristics of various filter designs. © 2008 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.2943217]

Subject terms: correlation filters; target detection; image analysis.

Paper 080009R received Jan. 8, 2008; revised manuscript received Mar. 28, 2008; accepted for publication Apr. 1, 2008; published online Jun. 23, 2008.

1 Introduction

The field of correlation-filter-based pattern recognition has been researched extensively over the past three decades and successfully used in various applications.¹⁻⁵ This success is largely due to attractive properties such as shift invariance, graceful degradation, and closed-form solutions. The simplest correlation filter (CF) is the matched filter, in which the filter template is matched to a single training image. It is well known that the matched filter is optimal for detection of a known object in additive white noise; however, the use of matched filters is not attractive for automatic target recognition (ATR), since the number of matched filters needed is very large when a wide range of target distortions must be handled.

In contrast to simple matched filters, composite CF designs (e.g., Refs. 6-14) allow the use of multiple training images and can produce a filter template that tolerates different types of noise and distortion. Many such designs also optimize performance criteria such as peak sharpness, output similarity, and low output noise variance⁷ for improved performance in various conditions. Thus, when using CFs for a detection/recognition task in the presence of distortion, we will likely need fewer composite CFs than matched filters. Unfortunately, because so many composite filter designs have been proposed in the literature, it is often unclear which one(s) to choose in solving a particular problem.

In this paper, we provide a survey and comparison of composite filter designs. We first provide a theoretical comparison, in which we expose the design motivations, underlying theory, and computational demands of each design. We then conduct an empirical comparison, where we test the performance of each design on a common data set by

measuring the separation between true-class and false-class scores. Our primary goal is to aid in selecting a good filter design for a given application.

For testing the filters, we use images of three out-of-plane-rotated scale-model tanks. We use both the original images and versions with added white Gaussian noise (AWGN). We train the filters on two of the classes, where the filter recognizes one target class and rejects another; however, we test the filters on three different classes (one of which was not seen during training). By comparing performance on the images with and without noise, we can observe the noise tolerance of each filter design. We also vary the amount of distortion each filter is trained to handle, allowing us to compare what we refer to as the *capacity* of the various filter designs, that is, how much distortion a particular filter can be trained to tolerate with acceptable performance. The generalization capability of each design can be seen by observing results on the unseen target class. Observing the overall results shows us which filter designs are best performers on this type of target recognition problem.

In addition to our primary goal, we have several secondary motives for our filter comparison:

1. Because of the attractive computational properties of eigenfilter designs,¹⁵ we want to compare the performance of eigenfilters with that of standard filters, looking specifically at generalization properties (e.g., noise tolerance, performance on unseen targets).
2. We want to provide an empirical comparison of many popular filter designs, which to our knowledge has not appeared in the literature.
3. We want to look for insights into different filter design methodologies, e.g., comparing capacity and noise tolerance trends between constrained and unconstrained designs.

While we should avoid drawing general conclusions

about different filter designs based on a single data set, we believe that our empirical comparisons provide some interesting insights into distortion-tolerant CF design for ATR applications.

The rest of this paper is organized as follows. Section 2 explains our choice of filter designs included in this paper. Section 3 provides a review of fundamental CF theory. In Secs. 4–7, we review the theory of the included designs, which we have divided into four different classes: constrained linear, unconstrained linear, overconstrained linear, and nonlinear. In Sec. 8, we discuss some computational aspects of the included designs. In Sec. 9, we describe our experiments for comparing the filter designs, the results of which are provided in Sec. 10. Finally, we summarize our conclusions in Sec. 11.

2 Summary of Included Designs

The filter designs considered in this paper may be divided into several classes. We refer to the first class as *constrained linear* CFs; it includes such designs as OTSDF,⁹ MACE,⁷ and MINACE¹⁰ filters. Designs in this class have the property that the designer specifies (constrains) the correlation output value for every training image. This property is commonly used to achieve a distortion-tolerant filter by specifying a constant output value over a representative set of distorted training images. We describe several constrained linear CF designs in Sec. 4. Removing these hard constraints enlarges the filter solution space and leads to another class of filter designs that are commonly referred to as *unconstrained linear* filters. This class includes such designs as MACH¹¹ EMACH,¹² and UOTSDF¹⁶ filters. We describe these designs in Sec. 5.

Several linear filter designs have been proposed that have fewer degrees of freedom than the number of constraints. We refer to such designs as *overconstrained linear* filters. Some of these designs, such as eigenfilters,¹⁵ have a finite number of constraints, while others, such as the optimal trade-off circular harmonic function (OTCHF) filter,¹⁷ have an infinite number. All such designs constrain the filter to lie in a smaller subspace than that spanned by the training data. While this type of approach might seem counter-intuitive in light of the motivations for unconstrained filter designs (e.g., enlarged solution spaces), it has been shown to improve filter generalization,^{13,18} which can lead to better performance on unseen data. Because these filter design problems are overconstrained, the actual response of the filter to each training image will likely be different than the specified response; thus, filters in this class are designed by minimizing an error metric between the desired and actual responses. We describe several existing overconstrained designs and some proposed modifications in Sec. 6.

One of the main advantages of linear CFs is their inherent shift invariance, i.e., the convolution operation effectively projects the input pattern vector onto the filter vector at every shift. For this reason, the input pattern need not be centered beforehand. In general, this property is not present in nonlinear classifiers (e.g., neural networks, SVMs); consequently, if the center is unknown, the pattern must either undergo preprocessing to locate the center or be presented to the classifier at all plausible shifts. However, several nonlinear CF designs have been proposed that retain the shift invariance of linear filters, using specific implementa-

tion schemes. Two such classes of designs are quadratic CFs^{14,19} and polynomial CFs.^{20,21} We describe specific designs from these two classes in Sec. 7.

Often, the computational load is an important factor in solving a recognition problem. Depending on the nature of the design (e.g., constrained or unconstrained), different filter designs have varying computational demands, especially in the design stage. We compare this aspect of the different filter designs in Sec. 8.

In order to compare filter designs on an equal basis, we limit our focus to filter designs that meet the following two criteria: (1) the design accepts and uses any number of true-class training images (for general distortion tolerance), and (2) the final output is intended to be a single correlation array. These criteria allow for solving a broad class of ATR problems, since any set of training images may be included and the output can be thresholded for any number of detections. Many other correlation filtering schemes have been proposed that are well suited for certain target recognition problems but do not meet these criteria. Some noteworthy examples include lock-and-tumbler filters, proposed by Schils and Sweeney,^{22,23} which look for points of constancy among multiple outputs rather than maximum height, and linear phase coefficient composite filters (LPCCFs), proposed by Hassebrook et al.,²⁴ which select only those training images that yield a near-Toeplitz inner-product matrix in order to achieve rotation invariance. We realize that our survey may exclude some filter designs in the literature that meet the preceding criteria; however, we believe that we have included those filters that have received significant attention from the ATR community in recent years.

Among the other omitted designs are the distance classifier CF (DCCF)²⁵ and its nonlinear extension, the polynomial DCCF (PDCCF).²⁶ These designs have been shown to work well for synthetic aperture radar (SAR) target recognition when the input images are of single targets.²⁷ The DCCF and PDCCF designs transform the input images so that the target classes are maximally compact and separated with respect to a particular distance metric. While full correlations are carried out over the entire image, these correlation values are optimized for use in computing a single distance value rather than for array thresholding. Thus, the DCCF scheme would be more appropriate as a target recognition stage than as a detection stage.

Also of interest are the class of filters (Goudail and Réfrégier,²⁸ Seong and Choi²⁹) that assume that the target and background are nonoverlapping regions of noise with different statistics. These designs represent a very different approach to target detection, in that they do not make use of exemplary training images but instead assume a known target silhouette. In addition, the statistics of the random gray levels in the target and background regions are assumed to be known. While this approach may have value in many pattern recognition applications, it does not use the same type of training images as the other filter designs; therefore, we do not consider these filters in our comparison.

3 Review of Correlation-Filter Basics

Correlation filtering may be divided into two stages: the design (synthesis) stage and the use stage. It is important to distinguish between these two stages. The design stage of a CF is often computationally expensive; however, each filter

need only be synthesized once, and the filter thereafter can be used efficiently in the frequency domain.

Linear filters are typically used in the following manner. Let \mathbf{x} denote the vectorized version of the 2-D image $x[m, n]$ in the space domain, formed by lexicographically ordering the elements of the image. In the same way, let \mathbf{h} denote the vectorized version of the filter impulse response $h[m, n]$. The correlation value at the origin is equal to the dot product $\mathbf{h}^T \mathbf{x}$. The correlation output $c[m, n]$ at every shift may be computed by correlating the image with the filter:

$$c(m, n) = \sum_i \sum_j x[i, j] h[i - m, j - n]. \quad (1)$$

However, this output array is more efficiently computed in the frequency domain. If we let $X[k, l]$ and $H[k, l]$ denote the 2-D discrete Fourier transforms (DFTs) of $x[m, n]$ and $h[m, n]$, respectively, then $c[m, n]$ can be expressed as the inverse 2-D DFT of the frequency domain conjugate product as follows:

$$c[m, n] = \mathcal{F}^{-1}\{X[k, l] H^*[k, l]\}. \quad (2)$$

This array is sometimes referred to as the “correlation plane,” a term that originated in the field of optical filtering. Some nonlinear filter designs (e.g., quadratic filters) compute the correlation output using a different scheme than Eq. (2); nevertheless, the resulting array is treated in the same way, regardless of how it was computed.

Once the output array is computed, each value may be compared with a predetermined threshold for detections. In many ATR applications, however, correlation peaks are measured in terms of sharpness, i.e., relative height compared to surrounding values.⁵ This type of metric is useful for at least two reasons: (1) it achieves invariance to local and global variations in image brightness (especially for filters with a nonzero dc component), and (2) many filter designs attempt to maximize some measure of peak sharpness by controlling more than one value in the output. A commonly used sharpness metric is the peak-to-sidelobe ratio (PSR),³⁰ which we define here by the following equation:

$$\text{PSR} \triangleq \frac{\text{peak} - \mu}{\sigma}, \quad (3)$$

where μ and σ are the mean and the standard deviation, respectively, of the correlation values inside a small window surrounding the peak. A small area around the peak may be excluded from the window to avoid contributions from broad correlation peaks.

It should be noted that PSR is typically defined as the *absolute value* of the given expression, so that all resulting PSR values will be positive. When PSR is used to measure a correlation value whose height is well above the surrounding values (i.e., a true correlation peak), the presence or absence of the absolute value operation makes no difference. In this work, however, we often measure PSR at locations where the correlation value may not be a peak. Using our definition, PSR values may be positive or negative; thus, a positive PSR value indicates that the correla-

tion value is above the mean of the surrounding values, while a negative value indicates that it is below the mean.

We typically use the PSR metric in scenarios where targets are large enough and show enough detail to be distinguished by shape rather than by absolute brightness. The resulting PSR values represent a kind of normalized target likelihood metric. For example, a PSR value of 4 indicates that the peak is 4 standard deviations above the mean of the surrounding values, which is highly unlikely to be a random fluctuation, whereas a value of 1 or less suggests that the peak is likely to have been caused by noise or clutter rather than a target. Thus, PSR values, like normalized correlation values, can be consistently translated into statements of target likelihood, allowing them to serve as a useful component in a probabilistic framework.

While we use only the peak height and PSR metrics in this paper, other metrics may be useful for detecting correlation peaks. For example, the peak-to-correlation-energy (PCE) ratio,³¹ which measures the height of the peak relative to the energy in the entire correlation output, may be appropriate for applications in which it can be assumed that there is at most one target in the scene (e.g., face recognition).

PSR can be efficiently computed at every point in the correlation output using two convolutions (or four FFTs). This procedure is described here. Let $g(\mathbf{x})$ and $w(\mathbf{x})$ denote, respectively, the correlation output and a user-defined PSR window function, where $\mathbf{x} \in \mathcal{G} \triangleq \{1 \dots L\} \times \{1 \dots M\}$, i.e., \mathbf{x} takes discrete values on a 2-D grid of size $L \times M$ pixels. The window function w should be centered at the origin, and it should take the value $1/N_w$ inside the window and zero elsewhere, where N_w is the number of (nonzero) pixels in the window. Let W_x denote the set of points inside a window centered at point \mathbf{x} . Then we can compute the mean $\mu_w(\mathbf{x})$ of the values in the surrounding window of every point \mathbf{x} in the output via a simple convolution:

$$\mu_w(\mathbf{x}) = \frac{1}{N_w} \sum_{\mathbf{y} \in W_x} g(\mathbf{y}) = \sum_{\mathbf{y} \in \mathcal{G}} g(\mathbf{y}) w(\mathbf{x} - \mathbf{y}). \quad (4)$$

From Eq. (4) we define the function $F_{\text{mean}}[\cdot, \cdot]$ that takes as input two arrays (a correlation output and a window function) and outputs an array of local means as follows:

$$F_{\text{mean}}[g, w] \triangleq g * w, \quad (5)$$

where the operator $*$ denotes a 2-D discrete convolution.

Similarly, the standard deviations can be computed by convolving instead with the square of the correlation output:

$$\begin{aligned} \sigma_w(\mathbf{x}) &= \left\{ \frac{1}{N_w} \sum_{\mathbf{y} \in W_x} [g(\mathbf{y}) - \mu_w(\mathbf{x})]^2 \right\}^{1/2} \\ &= \left[\frac{1}{N_w} \sum_{\mathbf{y} \in W_x} g^2(\mathbf{y}) + \frac{1}{N_w} \sum_{\mathbf{y} \in W_x} \mu_w^2(\mathbf{x}) \right. \\ &\quad \left. - \frac{1}{N_w} \sum_{\mathbf{y} \in W_x} 2g(\mathbf{y})\mu_w(\mathbf{x}) \right]^{1/2} \\ &= \left\{ \left[\sum_{\mathbf{y} \in \mathcal{G}} g^2(\mathbf{y}) w(\mathbf{x} - \mathbf{y}) \right] - \mu_w^2(\mathbf{x}) \right\}^{1/2}. \end{aligned} \quad (6)$$

Table 1 Constrained linear filter designs and corresponding values of **T**.

Filter design	Value of T
ECP-SDF ⁶	I
MV-SDF ⁸	C
MACE ⁷	D
OTSDF ⁹	$\alpha \mathbf{C} + (1 - \alpha^2)^{1/2} \mathbf{D}$
MINACE ¹⁰	$\max(\alpha \mathbf{C}, (1 - \alpha^2)^{1/2} \mathbf{D}_1, \dots, (1 - \alpha^2)^{1/2} \mathbf{D}_N)$

From Eq. (6) we define another function $F_{\text{std}}[\cdot, \cdot]$ that takes the same inputs as F_{mean} and outputs an array of standard deviation values:

$$F_{\text{std}}[g, w] \triangleq \{(g^2 * w) - F_{\text{mean}}^2[g, w]\}^{1/2}. \quad (7)$$

From the two functions in Eqs. (5) and (7) we define a PSR function $F_{\text{PSR}}[\cdot, \cdot]$ according to the definition in Eq. (3) as follows:

$$F_{\text{PSR}}[g, w] \triangleq \frac{g - F_{\text{mean}}[g, w]}{F_{\text{std}}[g, w]}. \quad (8)$$

In all experiments in this paper, we applied the PSR metric to every correlation output per Eq. (8) before thresholding for detections.

4 Constrained Linear Filters

Many CF designs require that the filter designer specify the output of the filter for each training image provided. For N training images, this results in N constraints, which is typically much less than the number of free parameters, d (the dimensionality of the filter). For this reason, many of these designs optimize some filter performance criterion while satisfying the N constraints. We refer to this class of designs as *constrained linear* CFs. The general form of a constrained linear filter **h** is

$$\mathbf{h} = \mathbf{T}^{-1} \mathbf{X} (\mathbf{X}^T \mathbf{T}^{-1} \mathbf{X})^{-1} \mathbf{u}, \quad (9)$$

where **X** is a matrix whose N columns are the N vectorized frequency-domain training images, **T** is a diagonal preprocessor matrix (explained below), and **u** is an $N \times 1$ vector of the specified correlation output values for each training image.

Special cases of the preprocessor **T** result in well-known filter designs. These cases are listed in Table 1, where **C** and **D_i** are diagonal matrices containing the power spectral density (PSD) of the noise and the power spectrum of the i 'th training image, respectively, and **D** = $(1/N) \sum_{i=1}^N \mathbf{D}_i$ contains the average training power spectrum. The *optimal trade-off synthetic discriminant function* (OTSDF) filter⁹ includes a trade-off parameter α that allows the user to emphasize low output noise variance (ONV) (α closer to 1) or low average correlation energy (ACE) (α closer to 0). Setting $\alpha=1$ yields the *minimum-variance synthetic dis-*

criminant function (MV-SDF) filter,⁸ which has minimum ONV but usually exhibits broad correlation peaks; in contrast, setting $\alpha=0$ yields the *minimum-average correlation energy* (MACE) filter,⁷ which has minimum ACE and produces sharp peaks on the training images but is highly sensitive to noise and distortion. The *equal correlation peak synthetic discriminant function* (ECP-SDF) filter,⁶ the predecessor to the other filter designs, is a special case of the MV-SDF filter when white noise is assumed (i.e., **C**=**I**).

The *minimum noise and correlation energy* (MINACE) filter¹⁰ achieves an alternative compromise between these two extremes by using an envelope equal to the greater of the noise and training image power spectra at each frequency. It should be noted that the trade-off parameter α appearing in the MINACE formulation in Table 1 is not part of the traditional MINACE filter design as reported in Ref. 10; rather, the value of **C** is varied directly, since the input noise level is typically unknown. This difference is merely semantic; in practice, the same effect is achieved by varying either **C** or α . In both the OTSDF and the MINACE designs, because both the input noise level and the trade-off can be effected by scaling **C** relative to **D**, a single parameter α simultaneously accomplishes both of these goals.

5 Unconstrained Linear Filters

Because the outputs of constrained CFs on nontraining images are almost always different than those on training images, the value of enforcing hard constraints on training images in filter design has been questioned.¹¹ Relaxing or removing such constraints might lead to a larger filter solution space. Also, the matrix inversion in the constrained design may be ill-conditioned when highly similar training images are included. For these reasons, several *unconstrained linear* filter designs have been proposed. These designs maximize some measure of the average output on true-class training images while minimizing other criteria such as ONV and ACE.

The maximum average correlation height (MACH) filter¹¹ is one such design, which achieves distortion tolerance by maximizing the similarity of the shapes of true-class correlation outputs over the training images. This maximization is realized by minimizing a *dissimilarity* metric known as the *average similarity measure* (ASM). This design also attempts to simultaneously maximize the average correlation-peak height (ACH) of the true-class training images and minimize the ACE of the false-class images. The unconstrained OTSDF (UOTSDF) filter¹⁶ is a similar design that minimizes a trade-off between true-class ACE and ONV (as in the OTSDF design). These filters rely heavily on the average training image **m** (see the filter equations below), which may resemble clutter more than a target and lead to poor clutter rejection capability.

The extended MACH (EMACH) filter design¹² addresses this clutter rejection problem by reducing the dependence on the average training image **m**. A tunable parameter β [in Eq. (11) below] is used to control this reduction. Two new metrics are used in the design: (1) the all-image correlation height (AICH), which takes into account the filter output on **m** as well as on individual training images, and (2) a modified average similarity measure (MASM), which measures the average dissimilarity to the

Table 2 Unconstrained linear filter designs. Note that the EMACH and EEMACH filters differ only in the matrices \mathbf{C}_x^β and $\hat{\mathbf{C}}_x^\beta$; also, \mathbf{S}_x^0 comes from substituting $\beta=0$ in Eq. (10).

Filter design	Filter h
UOTSDF ¹⁶	$(\mathbf{D}_x)^{-1}\mathbf{m}$
MACH ¹¹	$[\alpha\mathbf{D}_y + (1-\alpha^2)^{1/2}\mathbf{S}_x^0]^{-1}\mathbf{m}$
EMACH ¹²	Dominant eigenvector of $[\alpha\mathbf{I} + (1-\alpha^2)^{1/2}\mathbf{S}_x^\beta]^{-1}\mathbf{C}_x^\beta$
EEMACH ¹⁸	Dominant eigenvector of $[\alpha\mathbf{I} + (1-\alpha^2)^{1/2}\mathbf{S}_x^\beta]^{-1}\hat{\mathbf{C}}_x^\beta$

optimal output shape. This optimal shape takes into account the reduced dependence on \mathbf{m} realized by the new AICH metric. The EMACH design also includes an ONV criterion to help maintain noise tolerance. A trade-off parameter α is used to control the relative importance of the ONV and MASM criteria, where higher values of α correspond to greater emphasis on ONV and vice versa.

Let \mathbf{x}_i , $i=1, \dots, N_x$, and \mathbf{y}_i , $i=1, \dots, N_y$, denote the vectorized frequency-domain true-class and false-class training images, respectively. Also, let \mathbf{D}_x and \mathbf{D}_y denote diagonal matrices containing the average power spectra of the true and false classes, and let \mathbf{C} denote the noise PSD matrix defined as in Sec. 4. If we define the matrices

$$\mathbf{S}_x^\beta = \frac{1}{N_x} \sum_{i=1}^{N_x} [\mathbf{X}_i - (1-\beta)\mathbf{M}][\mathbf{X}_i - (1-\beta)\mathbf{M}]^*, \quad (10)$$

$$\mathbf{C}_x^\beta = \frac{1}{N_x} \sum_{i=1}^{N_x} (\mathbf{x}_i - \beta\mathbf{m})(\mathbf{x}_i - \beta\mathbf{m})^T, \quad (11)$$

where \mathbf{X}_i and \mathbf{M} are diagonal matrices containing the vectors \mathbf{x}_i and \mathbf{m} , respectively, along their diagonals, then the UOTSDF, MACH, and EMACH filter solutions are given by the equations in Table 2.

It is interesting to note that the EMACH and UOTSDF filter designs do not make use of false-class training images (i.e., requiring the filter to produce a small output value for these images), in contrast to the MACH filter design, which utilizes the average power spectrum of the false class in the optimization criteria. Other versions of the MACH filter have been reported (e.g., Refs. 11 and 12) that do not make use of false-class images but include other criteria such as ONV. Vijaya Kumar et al.¹² compared the performance of EMACH filters with that of MACH filters of this latter type, i.e., no false-class images were used in either design. Using this type of MACH filter design, they showed that, for well-chosen values of β , EMACH filters exhibited significantly better recognition performance than MACH filters on a database of synthesized synthetic aperture radar (SAR) imagery. This imagery was synthesized using radar prediction software called Xpatch,³² which simulates SAR images of targets at specified view angles based on CAD

models. To our knowledge, no such comparison has been reported in the literature that employs false-class images in MACH filter training.

If we approximate the covariance matrix \mathbf{C}_x^β by only its dominant eigenvectors and eigenvalues to yield a new matrix $\hat{\mathbf{C}}_x^\beta$, the resulting filter solution (given in Table 2) is referred to as the eigenextended MACH (EEMACH) filter.¹⁸ The motivation behind the EEMACH design is that using only a few eigenvectors to approximate \mathbf{C}_x^β will cause the resulting filter to be less specific to the training images, thus having better generalization and capacity. Vijaya Kumar and Alkanhal¹⁸ showed that an EEMACH filter using only a single-eigenvector approximation outperformed MACH (with no false-class training) and EMACH filters for target recognition on the MSTAR³³ SAR image database. This somewhat surprising result suggests that using eigen-based approximations may be advantageous in other filter designs; this topic is further explored in Sec. 6.2.

6 Overconstrained Linear Filters

In constrained composite filter design problems, the number of hard constraints (i.e., the number of input images) is typically much less than the number of free parameters (i.e., the number of pixels in the filter). It has been hypothesized, however, that constraining the filter to lie in a subspace of the span of the training images will decrease the specificity of the filter to the training set and may lead to better generalization and performance.^{15,18} In this case, the number of free parameters of the filter will be equal to the dimensionality of the subspace. If we then specify a peak constraint for each training image, the number of constraints may exceed the number of free parameters, and the resulting filter design, which we refer to as *overconstrained*, solves for the filter that best approximates these constraints. We discuss several interesting cases of this class of filter designs.

6.1 Continuous-Response Filters

There exist certain types of distortions, including in-plane rotation and scale, for which subspaces of the training images can be represented algebraically by harmonic functions. In the case of in-plane rotation, these are the circular harmonic functions (CHFs), whereas for scale they are the logarithmic radial harmonic functions (LRHFs) and Mellin radial harmonic functions (MRHFs). These harmonic representations have led to several filter designs^{34,35} that achieve tolerance or invariance to the corresponding distortion type. Two recent designs are optimal trade-off circular harmonic function (OTCHF) filters¹⁷ and minimum-average-correlation-energy Mellin radial harmonic (MACE MRH) filters.³⁶ We refer to these designs as *continuous-response* filters.

The OTCHF filter design¹⁷ allows the user to specify a continuum of peak constraints for all in-plane rotated versions of a single training image. Specifically, the user provides a response function defined on the interval $[0, 2\pi)$, where each value in the domain corresponds to the angle of in-plane rotation, and each value in the range corresponds to the desired correlation peak on the training image at that angle. A typical response function is equal to one over some angular interval and zero everywhere else, which re-

sults in a rotation-tolerant filter with respect to the corresponding interval. The OTCHF filter design approximates this response function while minimizing the ACE criterion. It can be shown that this ACE minimization is equivalent to prewhitening the training image according to its average power spectrum across all rotations. The methodology of the MACH MRH filter design is similar to that of the OTCHF design, except that scale is considered rather than rotation; we refer the interested reader to the literature^{17,36} for details.

Hilai and Rubinstein³⁷ showed that if we formulate the (infinite) set of all in-plane rotated versions of a particular image, then the Karhunen-Loeve decomposition of this set will yield the circular harmonic functions of the image. This result implies that if we were to form a training set by generating many rotated versions of a prewhitened training image, we would be able to approximate the OTCHF filter via eigendecomposition of this training set. Therefore, it was hypothesized that both the OTCHF and MACE MRH designs could be replaced by a more general over-constrained design that accepts any representative set of training images. This hypothesis motivates the discussion of eigenfilters in the following subsection. It was shown³⁸ that eigenfilters outperformed both of these designs on sets of scaled and rotated tank images; therefore, we omit OTCHF and MACE MRH designs from our empirical comparisons. These designs still have value, however, in that they may be more analytically tractable than eigenfilters for performance predictions and design extensions.

6.2 Eigenfilters

The OTCHF and MACE MRH designs are based on algebraic descriptions of a particular type of distortion; for example, in-plane rotation can be described as a shift of the angular coordinates. Thus, parts of the methodology of these designs (e.g., continuous response functions) do not readily extend to other types of distortion that lack such mathematical properties. However, the idea of designing filters in subspaces is more general and can be applied to any type of distortion. Shenoy has claimed^{13,15} that such an approach should lead to filters with improved generalization and increased capacity.

In this section, we examine an existing subspace approach and propose several modifications. Shenoy¹⁵ has proposed two different subspace approaches; one (Ref. 15, Chap. 4) is designed to produce one filter per class (or macro-class), while the other (Ref. 15, Chap. 3) produces pairs of filters for discriminating between true- and false-class targets. In keeping with the filter design criteria described in Sec. 1, we focus on the former approach. We first describe two methods for choosing a subspace; we then show how to design a filter in a given subspace. We refer to this design approach as *eigenfilters*. We assume that the dimensionality N_e of the subspace has been predetermined.

6.2.1 Standard eigenfilter subspace

We use the term *standard eigenfilter* to refer to a general subspace approach, a case of which has been proposed by Shenoy (Ref. 15, Chap. 4) (the differences are explained at the end of Sec. 6.2.2). In this approach, all training images are first prewhitened (in the frequency domain) using a matrix \mathbf{T} selected by the user from the list in Table 1. This

matrix is computed using the average power spectrum over true- and false-class training images as described in Sec. 4. Thus, if \mathbf{x} is a vectorized frequency-domain image, then the prewhitened frequency-domain image is computed as

$$\mathbf{y} = \mathbf{T}^{-1/2} \mathbf{x}. \quad (12)$$

Let \mathbf{X} be a matrix of size $d \times N$ whose columns are the N vectorized frequency-domain training images, and let $\mathbf{Y} = \mathbf{T}^{-1/2} \mathbf{X}$ be the set of prewhitened training images. Also, let \mathbf{u} be a vector of length N that specifies the desired filter response to each training image.

We apply the Karhunen-Loeve (KL) transform³⁹ to this new data set to form an orthonormal basis as follows. We first compute a matrix \mathbf{V} whose columns are the normalized eigenvectors ϕ_i of length N , $i=1, \dots, N$, of the inner-product matrix $\mathbf{Y}^T \mathbf{Y}$. For each eigenvector ϕ_i , we compute a value $p_i = \phi_i^T \mathbf{u}$. We then compute the matrix $\mathbf{E} = \mathbf{Y} \mathbf{V}$, the columns of which are vectorized images \mathbf{e}_i of length d , $i=1, \dots, N$, where $\mathbf{e}_i = \mathbf{Y} \phi_i$. These vectors \mathbf{e}_i are actually the unnormalized eigenvectors of the *outer-product* matrix $\mathbf{Y} \mathbf{Y}^T$ (of size $d \times d$), but were computed using the *inner-product* matrix $\mathbf{Y}^T \mathbf{Y}$ (of size $N \times N$), which typically requires much less computation, since we usually have that $N \ll d$. We refer to \mathbf{e}_i as the *eigenimages* of the prewhitened training set, since they are of the same dimensionality as the training images (as opposed to the eigenvectors ϕ_i of length N). Finally, we choose the basis of the filter design subspace to be the N_e eigenimages \mathbf{e}_i corresponding to the largest p_i^2 values. The purpose of this ordering is explained in Sec. 6.2.2.

Because the eigenimages in this approach are computed from a macro training set of true- and false-class images, they may fail to capture important differences between the two classes for discrimination purposes. For this reason, it was proposed³⁸ to modify the filter design subspace by adding one or more discriminative basis vectors to the set of eigenvectors. The HDGD algorithm, proposed by Casasent and Chen,⁴⁰ was used to find suitable vectors for this purpose. This algorithm finds those eigenvectors that provide good separation of the two class means and have low sensitivity to additive white Gaussian noise (AWGN). It was shown,³⁸ however, that the standard eigenfilter approach tends to include the very same eigenvectors as those selected by HDGD; thus, the proposed modification is likely unnecessary in most cases and is not included in our experiments in this paper.

6.2.2 Subspace filter design

Once the filter design subspace has been determined, the filter design proceeds as follows. Let the $d \times N_e$ matrix \mathbf{E} denote the subspace basis set, where the columns \mathbf{e}_i of \mathbf{E} are the individual basis vectors, i.e., the vectorized eigenimages selected via one of the preceding methods. Because the filter is constrained to be in the subspace spanned by \mathbf{e}_i , it must be a linear combination of these vectors:

$$\mathbf{h} = \sum_{i=1}^{N_e} a_i \mathbf{e}_i = \mathbf{E} \mathbf{a}, \quad (13)$$

where \mathbf{h} is the vectorized filter in the frequency domain and $\mathbf{a} \triangleq [a_1 \ a_2 \ \dots \ a_{N_e}]$. Let \mathbf{u} be a vector of length N that speci-

fies the desired filter response to each training image as before, and let $\tilde{\mathbf{u}}$ denote the actual filter response to each prewhitened image after synthesis. Then we can write that

$$\tilde{\mathbf{u}} = \mathbf{Y}^T \mathbf{h} = \mathbf{Y}^T \mathbf{E} \mathbf{a}. \quad (14)$$

We solve for the weight vector \mathbf{a}_{MMSE} that minimizes the mean squared error (MSE) $|\tilde{\mathbf{u}} - \mathbf{u}|^2$ between the desired and actual filter responses. The problem may be written as follows:

$$\mathbf{a}_{\text{MMSE}} = \arg \min_{\mathbf{a}} (\mathbf{Y}^T \mathbf{E} \mathbf{a} - \mathbf{u})^T (\mathbf{Y}^T \mathbf{E} \mathbf{a} - \mathbf{u}). \quad (15)$$

By taking the derivative of the expression in Eq. (15) and setting it equal to zero, the solution to this problem can be shown to be

$$\mathbf{a}_{\text{MMSE}} = (\mathbf{E}^T \mathbf{Y} \mathbf{Y}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{Y} \mathbf{u}. \quad (16)$$

Furthermore, as the solution in Eq. (16) and the corresponding minimized MSE value depend on the eigenvectors chosen for the matrix \mathbf{E} , it can be shown¹⁵ that choosing those eigenvectors with the largest p_i^2 values will result in the minimum MSE for this solution.

Finally, because we have assumed thus far that the input images to the filter have been prewhitened whereas in testing they will not have been, the final filter must be cascaded with a prewhitening filter, which can be implemented as a single linear filter by substituting Eq. (16) into Eq. (13) and multiplying by $\mathbf{T}^{-1/2}$ as follows:

$$\mathbf{h} = \mathbf{T}^{-1/2} \mathbf{E} (\mathbf{E}^T \mathbf{Y} \mathbf{Y}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{Y} \mathbf{u}. \quad (17)$$

Our proposed eigenfilter method generalizes from that of Shenoy (Ref. 15, Chap. 4) in the following respects. First, our method allows the use of any prewhitener \mathbf{T} , while Ref. 15 uses only the MINACE prewhitener. Second, Ref. 15 assumes that \mathbf{u} is a vector of all ones, i.e., only true-class training images are considered. We, on the other hand, place no restrictions on \mathbf{u} and therefore allow training on both true- and false-class images. Shenoy¹⁵ proves that the first eigenvector chosen by the p_i^2 metric will always correspond to the highest eigenvalue; however, when the restriction on \mathbf{u} is relaxed, this relationship is no longer true.

7 Nonlinear Correlation Filters

A linear correlation output is an array of scalar output values from a linear discriminant applied to the input image at every shift. While limited in capability by their linear nature, linear CFs have the important advantage of efficient frequency-domain computation; however, special cases of nonlinear discriminant functions have been proposed for which the attractive computational properties of linear filters may be retained by specialized implementation schemes. We refer to such systems as *nonlinear correlation filters*.

In this section we describe two classes of nonlinear CF designs. The first class of designs, quadratic correlation filters (QCFs), is characterized by solving for a quadratic discriminant function in d -dimensional space, where d is the number of pixels in the image. This quadratic discriminant can then be efficiently implemented as a set of linear filters via eigendecomposition. In contrast, the second class of

designs, polynomial correlation filters (PCFs), are sets of linear filters applied to multichannel input images, whose outputs are subsequently summed to form a single output. Both the QCF and PCF designs solve for the set of linear filters jointly in order to optimize performance criteria.

7.1 Quadratic Correlation Filters

If a linear correlation filter is said to implement a linear discriminant in a d -dimensional space, then a QCF implements a quadratic discriminant in that space. The output y of a QCF \mathbf{Q} at the origin for vectorized input image \mathbf{x} is computed according to the following equation:

$$y = \mathbf{x}^T \mathbf{Q} \mathbf{x}, \quad (18)$$

where the filter \mathbf{Q} is a matrix of size $d \times d$. It is important to note that this equation computes only the correlation value at the origin, i.e., the correlation at zero shift. Efficient computation of the entire correlation output array is discussed in Sec. 7.1.3 below.

Mahalanobis et al. have proposed several methods for computing the matrix \mathbf{Q} . We focus on two such methods: subspace quadratic synthetic discriminant functions (SSQSDFs)¹⁹ and Rayleigh quotient quadratic correlation filters (RQQCFs).¹⁴ These two methods were reported to perform best among the proposed quadratic filter designs on a particular data set. Other designs include minimum-variance quadratic synthetic discriminant functions (MVQSDFs)¹⁹ and QCFs based on the Fukunaga-Koontz transform.¹⁴

QCF designs assume that the training images are divided into two classes, the true (target) class (class 1) and the false (clutter) class (class 2). Unlike other filter designs, however, the QCF design procedure operates on space-domain images rather than frequency-domain images. Quantities such as ACE and ASM, which are easily expressed in the frequency domain, are not utilized in the QCF designs, and thus the correlation output energy can only be controlled by including shifted versions of training images in the false class. We observe, however, that in linear filter designs such as MACE, minimizing ACE is achieved by simply prewhitening the training images; this prewhitening was used in the eigenfilter design (Sec. 6.2.2) and typically has the effect of emphasizing the higher frequencies in the training images, which lead to sharper correlation peaks.

We elect to use training image prewhitening instead of shifted training images for our experiments. Two reasons for this decision are the following: (1) it is unclear how many shifted versions to include and at which shifts; and (2) using OTSDF prewhitening gives the designer a tunable parameter to trade off between ACE and ONV, which is simpler and more intuitive than adding various shifted images. Thus, we assume that each training image and input image is prewhitened by the OTSDF pre-whitener matrix \mathbf{T} given in Table 1. We explicitly prewhiten the training images before filter design via the method described by Eq. (12) in Sec. 6.2.1, and we show in Sec. 7.1.3 below how to incorporate the complementary prewhitening for input images into the QCFs themselves.

7.1.1 SSQSDF design

The SSQSDF design is a constrained design that assumes the filter \mathbf{Q} is contained in the subspace spanned by the outer products of the training images. More specifically, if we let \mathbf{x}_i denote the i 'th vectorized space-domain training image, then \mathbf{Q} is assumed to be of the form

$$\mathbf{Q} = \sum_{i=1}^{N_T} \beta_i \mathbf{x}_i \mathbf{x}_i^T, \quad (19)$$

where N_T is the number of training images. Let N_1 and N_2 be the numbers of class-1 and class-2 training images, respectively, where $N_1 + N_2 = N_T$, and let the training set be partitioned so that \mathbf{x}_i is a class-1 training image for $1 \leq i \leq N_1$ and a class-2 training image for $N_1 + 1 \leq i \leq N_T$.

Designing the filter is reduced to solving for the weights β_i . The filter is designed to satisfy $N_T + 2$ constraints, two on the output means and N_T on the output variances. The mean constraints are that $E_1[\mathbf{x}^T \mathbf{Q} \mathbf{x}] = 1$ and $E_2[\mathbf{x}^T \mathbf{Q} \mathbf{x}] = -1$, where E_1 and E_2 denote expectation with respect to training samples from class 1 (target class) and class 2 (clutter class), respectively. The variance constraints are $\mathbf{x}_i^T \mathbf{Q} \mathbf{x}_i - E_1[\mathbf{x}^T \mathbf{Q} \mathbf{x}] = 0$ for $1 \leq i \leq N_1$ and $\mathbf{x}_i^T \mathbf{Q} \mathbf{x}_i - E_2[\mathbf{x}^T \mathbf{Q} \mathbf{x}] = 0$ for $N_1 + 1 \leq i \leq N_T$. These latter constraints are similar to constraining the true-class images to produce an output value of 1 and the false-class images to produce an output of 0; however, since the mean constraints may not be satisfied exactly, the values may not be precisely 1 and 0.

Muise et al. derive a least-squares solution for the weights as follows. Let $\mathbf{R}_1 = E_1[\mathbf{x} \mathbf{x}^T]$ and $\mathbf{R}_2 = E_2[\mathbf{x} \mathbf{x}^T]$ denote the correlation matrices for each class. Then the matrix \mathbf{B} is formed as

$$\mathbf{B} = [\mathbf{p} \ \mathbf{q} \ \mathbf{p}_1 \ \cdots \ \mathbf{p}_{N_T}]^T, \quad (20)$$

where the vectors comprising \mathbf{B} are of length N_T and are defined as follows:

$$\mathbf{p} = [\mathbf{x}_1^T \mathbf{R}_1 \mathbf{x}_1 \ \cdots \ \mathbf{x}_{N_T}^T \mathbf{R}_1 \mathbf{x}_{N_T}]^T,$$

$$\mathbf{q} = [\mathbf{x}_1^T \mathbf{R}_2 \mathbf{x}_1 \ \cdots \ \mathbf{x}_{N_T}^T \mathbf{R}_2 \mathbf{x}_{N_T}]^T,$$

$$\mathbf{p}_i = \begin{cases} \begin{bmatrix} \mathbf{x}_1^T (\mathbf{x}_i \mathbf{x}_i^T - \mathbf{R}_1) \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{N_T}^T (\mathbf{x}_i \mathbf{x}_i^T - \mathbf{R}_1) \mathbf{x}_{N_T} \end{bmatrix}, & 1 \leq i \leq N_1, \\ \begin{bmatrix} \mathbf{x}_1^T (\mathbf{x}_i \mathbf{x}_i^T - \mathbf{R}_2) \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{N_T}^T (\mathbf{x}_i \mathbf{x}_i^T - \mathbf{R}_2) \mathbf{x}_{N_T} \end{bmatrix}, & N_1 + 1 \leq i \leq N_T. \end{cases} \quad (21)$$

It can be shown that the constraint equation is then equivalent to

$$\mathbf{B} \boldsymbol{\beta} = \mathbf{c}, \quad (22)$$

where we define

$$\mathbf{c} = [1 \ -1 \ 0 \ \cdots \ 0]^T,$$

$$\boldsymbol{\beta} = [\beta_1 \ \cdots \ \beta_{N_T}]^T. \quad (23)$$

The least-squares solution for $\boldsymbol{\beta}$ in Eq. (22) is given by

$$\boldsymbol{\beta} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{c}, \quad (24)$$

which yields the filter \mathbf{Q} when substituted into Eq. (19).

7.1.2 RQCCF design

The RQCCF design is an unconstrained design that attempts to maximize the objective function $J(\mathbf{x}) = E_1[\mathbf{x}^T \mathbf{Q} \mathbf{x}] - E_2[\mathbf{x}^T \mathbf{Q} \mathbf{x}]$, where E_1 and E_2 are as previously defined. This effectively maximizes a measure of separation between the filter responses of the two classes. This design assumes that the filter is of the form

$$\mathbf{Q} = \sum_{i=1}^{N_1} \gamma_i \mathbf{q}_i \mathbf{q}_i^T - \sum_{i=1}^{N_2} \delta_i \mathbf{p}_i \mathbf{p}_i^T, \quad (25)$$

where \mathbf{q}_i , $1 \leq i \leq N_1$, and \mathbf{p}_i , $1 \leq i \leq N_2$, are a set of orthonormal vectors (not necessarily the training images as in SSQSDF). It can be shown¹⁴ that the objective function $J(\mathbf{x})$ may be made large by choosing sets \mathbf{q}_i and \mathbf{p}_i to be the first N_1 and N_2 eigenvectors of $\mathbf{R}_1 - \mathbf{R}_2$ and $\mathbf{R}_2 - \mathbf{R}_1$, respectively, corresponding to the largest eigenvalues. Because these two sets of eigenvectors and eigenvalues will simply be the negatives of each other, this selection scheme results in choosing the N_1 eigenvectors of $\mathbf{R}_1 - \mathbf{R}_2$ with the most positive eigenvalues for \mathbf{q}_i and the N_2 eigenvectors of the same matrix with the most negative eigenvalues for \mathbf{p}_i .

Once these vectors are chosen, it remains only to find the RQCCF weights γ_i and δ_i . The approach of Ref. 14 is to choose

$$\gamma_i = \begin{cases} 1, & \mathbf{q}_i^T (\mathbf{R}_1 - \mathbf{R}_2) \mathbf{q}_i > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (26)$$

$$\delta_i = \begin{cases} 1, & \mathbf{p}_i^T (\mathbf{R}_2 - \mathbf{R}_1) \mathbf{p}_i > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

By substituting into Eq. (25), the combination of vectors \mathbf{q}_i and \mathbf{p}_i and their corresponding weights yield the QCF matrix \mathbf{Q} .

7.1.3 QCF implementation

Once the QCF matrix \mathbf{Q} is computed, a naive (and computationally inefficient) way to compute the entire quadratic correlation output, i.e., the correlation value at every shift of the input, would be to form every shifted version of the original input image into a vector \mathbf{x} and reapply Eq. (18). It can be shown, however, that if there exist matrices \mathbf{F} and \mathbf{G} of sizes $d \times N_F$ and $d \times N_G$ such that \mathbf{Q} can be written as

$$\mathbf{Q} = \mathbf{F} \mathbf{F}^T - \mathbf{G} \mathbf{G}^T, \quad (28)$$

then the output array can be computed using a series of linear convolutions. Specifically, if we reshape each column of \mathbf{F} and \mathbf{G} into an image of size $L \times M$, then the output array $y[l, m]$ is given by the following equation:

$$y[l, m] = \sum_{i=1}^{N_F} |x[l, m] \otimes f_i[l, m]|^2 - \sum_{i=1}^{N_G} |x[l, m] \otimes g_i[l, m]|^2, \quad (29)$$

where $x[l, m]$ is the input image, $f_i[l, m]$ and $g_i[l, m]$ are the images corresponding to the j 'th columns of \mathbf{F} and \mathbf{G} , and the operator \otimes denotes a 2-D discrete correlation.

In order to decompose a QCF matrix \mathbf{Q} into components \mathbf{F} and \mathbf{G} , let \mathbf{u}_i and λ_i , $1 \leq i \leq d$, denote the eigenvectors and corresponding eigenvalues of \mathbf{Q} , sorted by eigenvalues in decreasing order. Suppose we want to approximate \mathbf{Q} using N_1 positive eigenvalues and N_2 negative eigenvalues, which would result in $N_1 + N_2$ linear filters for use in the convolution formula in Eq. (29). If we define the matrices \mathbf{F} and \mathbf{G} as

$$\mathbf{F} = [\lambda_1^{1/2} \mathbf{u}_1 \quad \cdots \quad \lambda_{N_1}^{1/2} \mathbf{u}_{N_1}], \quad (30)$$

$$\mathbf{G} = [\lambda_{d-N_2}^{1/2} \mathbf{u}_{d-N_2} \quad \cdots \quad \lambda_d^{1/2} \mathbf{u}_d], \quad (31)$$

the $\mathbf{Q} \approx \mathbf{F}\mathbf{G}^T - \mathbf{G}\mathbf{G}^T$ as in Eq. (28), yielding a linear-filter approximation to the QCF. Once the matrices \mathbf{F} and \mathbf{G} are found, we incorporate the prewhitener \mathbf{T} into the linear filters contained in its columns to obtain new filter matrices \mathbf{F}' and \mathbf{G}' as follows:

$$\mathbf{F}' = \mathbf{T}^{-1/2} \mathbf{F}, \quad (32)$$

$$\mathbf{G}' = \mathbf{T}^{-1/2} \mathbf{G}. \quad (33)$$

The linear filters are thus implicitly cascaded with a prewhitener such that the assumption of prewhitened input images is satisfied.

7.2 Polynomial Correlation Filters

PCFs^{20,21} are co-designed sets of linear filters applied to multichannel images. These channels could be different nonlinear mappings of an image, images from different sensors, or a combination of both. The motivation for this design is that multiple channels may provide more information than a single channel for use in discrimination. We describe both a constrained and an unconstrained version of PCF design in this subsection.

Let us assume that each input to our filter consists of q channels. We denote each vectorized training input as

$$\mathbf{r}_i = \begin{bmatrix} \mathbf{x}_i^1 \\ \vdots \\ \mathbf{x}_i^q \end{bmatrix}, \quad (34)$$

where \mathbf{x}_i^j is the j 'th channel vectorized image of training input i . The final filter output c at the origin for a given input \mathbf{r} is computed from q linear filters $\mathbf{h}_1, \dots, \mathbf{h}_q$ by applying each filter to the corresponding channel image and summing the q outputs. Thus, the output can be expressed as $c = \mathbf{h}^T \mathbf{r}$, where

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_q \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_q \end{bmatrix}. \quad (35)$$

The PCF filter design solves simultaneously for the q filters. In order to optimize certain performance criteria, we define the following diagonal matrices (which have analogs in linear filter designs):

$$\mathbf{S}_{kl} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i^k (\mathbf{X}_i^l)^* - \mathbf{M}^k (\mathbf{M}^l)^*,$$

$$\mathbf{D}_{kl} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i^k (\mathbf{X}_i^l)^*, \quad (36)$$

where \mathbf{X}_i^k is a diagonal matrix with \mathbf{x}_i^k along the diagonal, and $\mathbf{M}^k = (1/N) \sum_{i=1}^N \mathbf{X}_i^k$. The diagonals of \mathbf{S}_{kl} and \mathbf{D}_{kl} are cross-spectral quantities used to formulate expressions for ACE and ASM, respectively. We further define diagonal matrices \mathbf{C}_{kl} , used in the ONV expression, each of whose diagonals contain the noise cross-spectral density between channels k and l . When $k=l$, each of these terms becomes a power-spectral quantity for the corresponding channel (rather than a cross-spectral quantity). If we define large matrices \mathbf{S} , \mathbf{D} , and \mathbf{C} of the form

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \cdots & \mathbf{S}_{1q} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \cdots & \mathbf{S}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{q1} & \mathbf{S}_{q2} & \cdots & \mathbf{S}_{qq} \end{bmatrix} \quad (37)$$

(where \mathbf{D} and \mathbf{C} are formed in the same way), then it can be shown^{21,41} that the filter performance criteria for the PCF may be expressed as follows:

$$\text{ASM} = \mathbf{h}^+ \mathbf{S} \mathbf{h},$$

$$\text{ACE} = \mathbf{h}^+ \mathbf{D} \mathbf{h},$$

$$\text{ONV} = \mathbf{h}^+ \mathbf{C} \mathbf{h}. \quad (38)$$

We present two PCF designs: a modified version of the unconstrained PCF (UPCF) proposed by Mahalanobis and Vijaya Kumar,²⁰ in which we replace the ASM criterion with ONV and ACE, and a variant of the constrained PCF (CPCF) proposed by Al-Mashouq et al.,²¹ in which we use the same trade-off. This latter variant of the PCF filter has been derived by Thornton.⁴¹

The constrained PCF design minimizes a weighted sum of ONV and ACE analogous to the OTSDF design (Sec. 4), using the trade-off parameter α in a similar manner. If we define $\mathbf{R} = [\mathbf{r}_1 \dots \mathbf{r}_q]$ and $\mathbf{T} = \alpha \mathbf{C} + (1 - \alpha^2)^{1/2} \mathbf{D}$, then the filter can be shown^{21,41} to have the following expression:

$$\mathbf{h}_{\text{CPCF}} = \mathbf{T}^{-1} \mathbf{R} (\mathbf{R}^T \mathbf{T}^{-1} \mathbf{R})^{-1} \mathbf{u}, \quad (39)$$

where \mathbf{u} is the peak constraint vector.

Table 3 Comparison of computational loads incurred in filter synthesis and use stages. Typical values for the parameters are $N=50$, $N_e=20$, $q=2$, $d=400$.

Filter	Design (synthesis)		Use FFTs
	Matrix inversions	SVDs (# of vectors)	
OTSDF and variants,	1 ($N \times N$)	0	2
MINACE			
MACH,	0	0	2
UOTSDF			
EMACH	0	1 of $N \times N$	2
EEMACH	0	1 of $N \times N$; N_e of $N \times N$	2
Eigen filters	1 ($N_e \times N_e$)	N_e of $N \times N$	2
CPCF	1($qd \times qd$)+1($N \times N$)	0	$q+1$
UPCF	1 ($qd \times qd$)	0	$q+1$
RQQCF	0	N_e of $d \times d$	$1+N_e$
SSQSDF	1 ($N \times N$)	N_e of $d \times d$	$1+N_e$

The unconstrained PCF design maximizes the average correlation height while minimizing a weighted sum of ONV and ACE as before, which can be shown²⁰ to yield the following equation for the filter:

$$\mathbf{h}_{\text{UPCF}} = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{m}^1 \\ \vdots \\ \mathbf{m}^q \end{bmatrix}, \quad (40)$$

where $\mathbf{m}^k = (1/N) \sum_{i=1}^N \mathbf{x}_i^k$ are the vectorized mean images of each channel. We observed in preliminary experiments that using the ONV-ACE trade-off resulted in superior performance to using the ASM criterion as proposed by Mahalanobis and Vijaya Kumar.²⁰

8 Computational Considerations

In this section, we describe the computational complexity of each filter design in both synthesis and use stages. Rather than quantifying complexity at the floating-point operation level, we provide a high-level comparison of computation in terms of three main quantities: (1) number of 2-D fast Fourier transformations (FFTs), (2) number and size of matrix inversions, and (3) number and size of singular value decomposition (SVD) operations, including how many eigenvectors need to be found. Our comparison is shown in Table 3. We use the following notation: N denotes the number of original training images (true-class plus false-class); N_e denotes the number of eigenimages to be used (in applicable designs); d is the dimensionality of the training images (i.e., the total number of pixels); and q

denotes the number of channels used in the PCF. We illuminate several important computational issues below.

Because quadratic filters are included in our comparison, it is important to consider the number of FFTs in the use stage. In determining this number, we consider that a given input image (or each channel of such for PCFs) must be transformed into the frequency domain before applying the filter, and that the filter-input product(s) (plural for QCFs) must be transformed back into the space domain. It is important to note that when PSR values are computed for each correlation output, the computational advantage or disadvantage (in the use stage) of using one or more nonlinear filters in place of several linear filters may be affected. Consider that computing PSR over the entire output, as described in Sec. 3, requires four additional FFTs. Each QCF will thus require N_e+5 FFTs, whereas each linear filter will require $2+4=6$ FFTs. For example, if two QCFs, each using three eigenvectors, could take the place of three linear filters, the QCF filter bank would require 16 FFTs as opposed to 18 FFTs for the linear filter bank. In some applications, a prescreening stage may filter out all but a few locations of interest, and thus PSR may only need to be computed at these specific locations. In this case, computing PSR at each point requires no FFTs, but instead is only a matter of computing the mean and standard deviation of the surrounding values.

The SSQSDF filter design incurs a heavy computational load in the design stage because the formulation of the vectors \mathbf{p}_i [Eq. (21)] requires many large matrix multiplications. The use stage of both of the quadratic filter designs requires more computation than that of linear filters, because quadratic correlation outputs are approximated by N_e linear filtering operations. In spite of these computational difficulties, however, a single QCF filter, due to its quadratic form, may be able to take the place of several linear filters in handling a given distortion and may thus provide a computational advantage in the use stage if N_e is sufficiently low.

The design stage of the PCF filters is computationally expensive due to the inversion of a matrix of size $qd \times qd$. Storing the matrix inverse as a whole may also be problematic; for example, if we use two image channels each of size 30×40 , the resulting matrix inverse will be of size 2400×2400 , or 5.76 million elements. The use stage requires that each channel image be transformed to the frequency domain (resulting in q FFTs), and only one additional FFT is required to generate the correlation output, since the q outputs may be summed in the frequency domain.

An important computational property of the eigenfilter designs is that repeated filter synthesis from the same training set does not require that the eigenvectors be recomputed. For example, suppose we compute an eigenspace for the entire 360-deg range of out-of-plane rotation with respect to some target, and suppose we then compute an eigenfilter with tolerance over the range $[0, 30 \text{ deg}]$ (which would require a small $N_e \times N_e$ matrix inversion as well). If we later decide that we need a filter for the range $[10, 40] \text{ deg}$, we can reuse the eigenspace from earlier and simply carry out another small-matrix inversion. Also, stor-

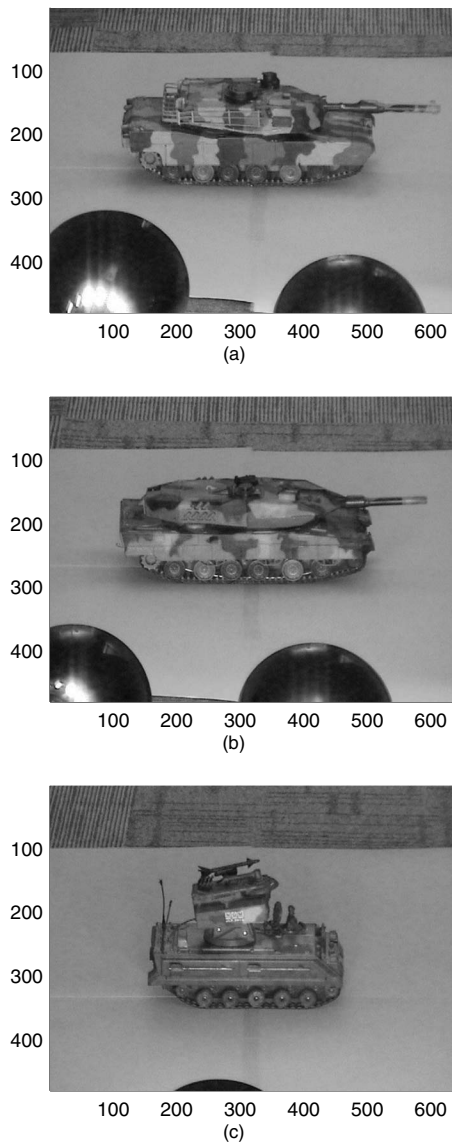


Fig. 1 Photograph of each model used in database acquisition: (a) Abrams, (b) Leopard, (c) Missile. The shades of the lights used for illumination are visible in the bottoms of the images.

ing the eigenspace eliminates the need to store the training images; thus, rather than storing N training images, we need only store the N_e ($<N$) eigenimages.

9 Experimental Setup

In this section, we describe the experimental setup of our empirical filter design comparisons. We describe the data used in our experiments in Sec. 9.1. We then describe our general method for measuring and comparing performance in Sec. 9.2. Finally, we describe our method for optimizing the various filter designs in Sec. 9.3.

9.1 Data

Our database consists of still images of out-of-plane-rotated scale-model vehicles. We used three different Marui scale models (acquired from <http://www.raidentech.com>) in our database acquisition. The first two models were 1/24-scale

Table 4 Elevation angles of each target included in different data sets.

Target	Elevation angle (deg)		
	Training set	Validation set	Testing set
Leopard	17	19	21
Abrams	17	19	21
Missile	23	19	21

tanks, one modeled after the U.S. M1A2 Abrams battle tank, and the other modeled after the German Leopard II A5 battle tank. Imagery of these two targets was captured at depression angles of 17, 19, and 21 deg. The third target was a missile-launching vehicle with no real-world equivalent, and rotational imagery of this target was captured at 19, 21, and 23 deg (17 deg was too shallow, due to the height of the target compared to the size of the background). We refer to the three models respectively as *Abrams*, *Leopard*, and *Missile*.

The dimensions of each scale-model tank were $41 \times 18 \times 13$ cm ($L \times W \times H$), while the dimensions of Missile were $30 \times 18 \times 15$ cm. All target images were captured at a resolution of approximately 0.4 cm/pixel. The turret of the missile launcher was set at an angle of approximately 16 deg above horizontal in all images. Full-resolution photographs of each target are shown in Fig. 1.

Each image in the database was generated from a photograph of the target against a bright green background atop a turntable. The scale-model targets were photographed from many sides, covering the entire 360-deg range of azimuth angles with an interval of approximately 1.5 deg. We automatically segmented the target from the background in each image, using morphological operations on the hue and intensity channels.

In creating each training image, a downsized version of the segmented target image (using the MATLAB function *imresize*,⁴² which performs lowpass filtering and bilinear interpolation) was placed in front of a constant background whose intensity was equal to the mean intensity of all target images from that class. These background values are listed at the end of Table 5 (Sec. 9.3). We measure the downsized target images in terms of pixels of horizontal target width (PHTW), that is, the distance in pixels from the leftmost to the rightmost target pixels. Examples of true- and false-class training images are shown in Fig. 2.

While it might seem that using different background values for each class could cause the filter to discriminate based on the background rather than the target pattern, any prewhitening (which is employed by all the filter designs) will alter the dc component of each image and cause the resulting backgrounds to be nonconstant and have varying means. For this reason, and because the dynamic range of the target patterns is significantly greater (≈ 5 –10 times) than the difference between the original background values, we do not expect this to be a problem.

Table 5 Parameters of filter training for parameter optimization.

Parameter	Filter 1 value	Filter 2 value	Filter 3 value	Filter 4 value
Trained true target	Missile			
Trained false target	Leopard			
Azimuth range (deg)	[120, 150]	[150, 180]	[180, 210]	[210, 240]
Scale range (PHTW)	[16, 20]			
No. of azimuth samples	$\begin{cases} 7 & \text{(Missile)} \\ 9 & \text{(Leopard)} \end{cases}$	$\begin{cases} 7 & \text{(Missile)} \\ 10 & \text{(Leopard)} \end{cases}$	$\begin{cases} 8 & \text{(Missile)} \\ 9 & \text{(Leopard)} \end{cases}$	$\begin{cases} 8 & \text{(Missile)} \\ 9 & \text{(Leopard)} \end{cases}$
No. of scale samples per azimuth sample	4			
Total no. of training images	64	68	68	68
Training-image background intensity	$\begin{cases} 69.3 & \text{(Missile)} \\ 79.4 & \text{(Leopard)} \end{cases}$			

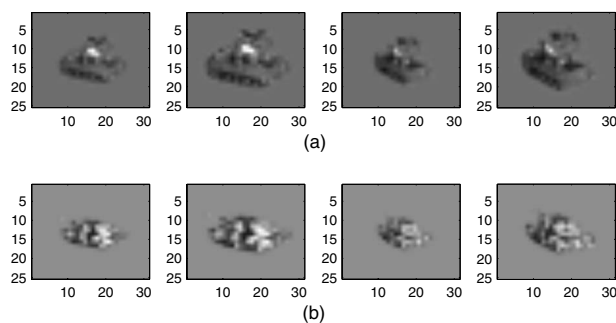
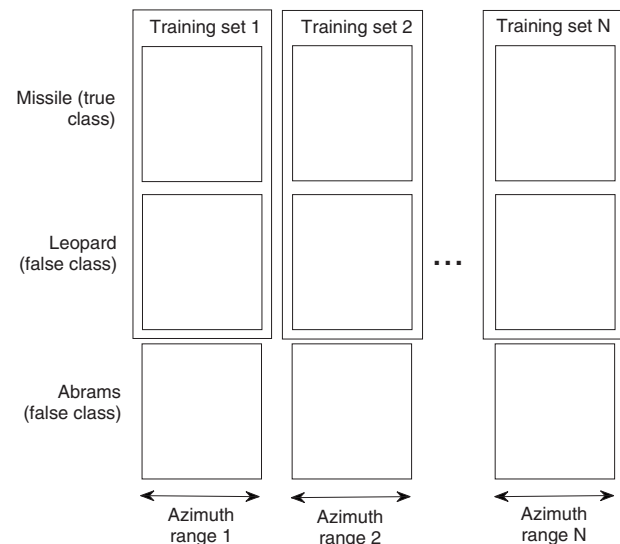
9.2 Filter Test General Setup and Metric

We divided our rotational database into three sets as follows (summarized in Table 4). First, we took as our *training* set all imagery of the Leopard and Abrams targets at a 17-deg elevation angle, and of the Missile target at a 23-deg elevation angle. These images only were used to train the filters. Second, we took as our *validation* set all Leopard, Abrams, and Missile imagery at a 19-deg elevation angle. These images were used as test images to select optimal filter parameters during synthesis (Sec. 9.3). Third, we took as our *testing* set all Leopard, Abrams, and Missile imagery at a 21-deg elevation angle. We used these images to test our synthesized filters and make performance observations (Sec. 10). We trained all filters using the Missile target as the true-class target.

Each filter was designed to handle a particular range of azimuthal target rotation. The true-class training images for each filter were taken only from within the filter's corresponding azimuth range; i.e., we did not include Missile images from other azimuth ranges as false-class images. In applicable designs, we trained each filter to reject training-set images of the Leopard target from the same azimuth

range as the true-class target. For example, in the constrained filter designs, we required the filter to produce a value of zero on these Leopard training images. We did not include images of the Abrams target directly in our filter synthesis; therefore, we refer to the Leopard and Abrams targets as *seen* and *unseen* false-class targets, respectively. It should be noted, however, that the unseen target class is used in selecting the optimal parameter(s) for each filter design. This training scheme is illustrated in Fig. 3.

We tested each filter on images from the validation set (i.e., at a different elevation angle) at two different noise levels: (1) no added noise, and (2) additive white Gaussian noise (AWGN) to yield a SNR of 20 dB. The AWGN was added to the entire image, and thus both the target and the background were affected by the noise. The SNR values

**Fig. 2** Examples of (a) Missile and (b) Leopard training images for parameter optimization.**Fig. 3** Illustration of filter training image selection scheme.

reported in this paper were obtained by measuring the energy of the target image before subtracting the mean (i.e., the background value). Because the mean value was significantly greater than zero, the reported value of 20 dB for the SNR is deceptively high. We observed that computing the SNR after mean subtraction resulted in a value of ≈ 3 dB, indicating that if we were to subtract the value of the background from the entire image, the resulting energy of the noise would be comparable to that of the target.

In order to quantify the discrimination ability of a particular filter bank at a given noise level, we compute the Fisher ratio (FR)³⁹ between all true-class and false-class PSR values from images with that noise level. This ratio is a measure of the separation between the two classes with respect to the filter outputs; thus, higher values correspond to better expected filter performance, assuming a well-chosen threshold. Specifically, if we define sets 1 and 2 to be the sets of filter responses on true-class and false-class images, respectively, then the FR is computed as

$$FR = \frac{|\mu_2 - \mu_1|^2}{\sigma_1^2 + \sigma_2^2}, \quad (41)$$

where μ_1 , μ_2 , σ_1^2 , and σ_2^2 are the means and variances of the two sets. We define a filter response to be the maximum PSR value (using an outer window of size 20×20 pixels) over an entire correlation output, where the output is computed via a circular convolution with the filter. The FR metric is often used for filter performance evaluation in ATR and other applications.^{43,44}

9.3 Filter Parameter Optimization

While some filter parameters (e.g., the input noise variance) may be known prior to filter design, it is often unclear how to choose appropriate values for trade-off parameters (e.g., between ACE and ONV). This problem is compounded when more than one parameter is available (e.g., EMACH, eigenfilters), in that exhaustive searches become infeasible over multidimensional spaces. We optimized the filter parameter(s) for each filter design independently, using a subset of the rotational imagery. We used these “optimal” parameter values for all further experiments, realizing that the determined values may not be optimal for different data.

For each filter design in our survey and each parameter value thereof, we synthesized a bank of four filters, each of which was trained to handle a separate range of azimuth angles but a common range of scale factors. For each filter bank, we computed two Fisher ratios for the resulting filter: one between the true class (Missile) and the seen false class (Leopard), and another between the true class (Missile) and the unseen false class (Abrams). For these parameter experiments, we trained on every second azimuthal angle in the training portion of the database, due to ill-conditioned matrix problems in the constrained designs when the training images were too similar to one another. Each filter was trained to tolerate a certain amount of scale distortion by artificially scaling each training image to four different sizes and including all such scaled images in the new training set. We used azimuthal ranges of [120, 150], [150, 180], [180, 210], and [210, 240] deg for the four filters, and for all filters we used a scale range of [16, 20] PHTW. These design parameters are summarized in Table 5.

It should be noted that a training-image selection scheme is often used (e.g., by Patnaik and Casasent⁴⁵) to decrease the number of training images used in filter synthesis. Use of such a scheme has been shown to increase filter generalization¹⁰ and may also eliminate the ill-conditioning problems mentioned. We did not use any such scheme in our filter synthesis; rather, we simply provided the filter with all available training images as already described. Also, aside from the inherent normalization carried out in the constrained filter designs (e.g., OTSDF, MINACE), we did not explicitly normalize the training images.

We used all validation-set images from all four azimuthal ranges for testing each filter (no images were skipped). Each filter was tested on true-class images within its azimuth range and on false-class images over all four azimuth ranges. In creating the testing images beforehand, each testing image was scaled to the same four sizes as used in filter training and placed against a constant background equal to the mean intensity of its class. This resulted in an average of 49 true-class and 568 false-class testing images per filter. In computing the FR, we combined all true-class responses over the four filters into one response set and all responses from each false-class target type into two false-class response sets (one for Leopard targets and the other for Abrams).

In selecting the best α value for each case, we chose the highest value of α among those yielding approximately equal performance, i.e., our choice of α attempted to maximize noise tolerance (resulting from the ONV term) without sacrificing much performance. Thus, for example, if α values of 10^{-4} and 10^{-2} resulted in approximately the same Fisher separation (visually equal in the plot), we would select $\alpha = 10^{-2}$. We observed in preliminary experiments that this selection method resulted in better performance on unseen data than choosing based on the absolute highest FR. Parameter values determined for each filter design are summarized in Table 6. Note that we used the OTSDF prewhitener in the eigenfilter design, and we henceforth refer to this design as eigen-OTSDF (EOTSDF).

10 Empirical Comparison Results

Having selected parameter values for each filter design in the preceding experiments, we test the performance of all the designs using various sizes of filter banks. We first compare performance of the four-filter banks optimized on the [120, 240]-deg range (as described in Sec. 9.3), and we illustrate the behavior of the different filters using box-and-whisker plots. We then examine the effect of varying the size of the inner PSR window (see Sec. 3) on these same filter banks. Finally, in order to examine filter performance on an equal-computation basis, we test new filter banks that were trained to span the entire 360-deg range of azimuthal rotation while limiting the number of FFTs allowed in the use stage for each bank. These filter banks were trained in the same way as the four-filter banks but with different azimuthal ranges for each filter (see Sec. 10.2). We use graphs of FRs to compare performance across the different designs for different false-class sets and noise levels. In all our comparison experiments in this section, we test on the testing set rather than the validation set (see Table 4 for corresponding elevation angles).

Table 6 Selected filter parameters after optimization procedure. The second-channel exponent (SCE) parameter applies only to the PCF filter designs.

Filter	No noise				AWGN			
	α	β	N_e	SCE	α	β	N_e	SCE
OTSDF	10^{-2}	—	—	—	$10^{-0.3}$	—	—	—
MINACE	$10^{-2.5}$	—	—	—	$10^{-1.4}$	—	—	—
MACH	10^{-2}	—	—	—	10^{-2}	—	—	—
UOTSDF	$10^{-1.6}$	—	—	—	$10^{-0.1}$	—	—	—
EMACH	10^{-2}	0	—	—	10^{-2}	0	—	—
EEMACH	10^{-2}	0	1	—	10^{-2}	0	1	—
EOTSDF	10^{-2}	—	7	—	$10^{-0.7}$	—	5	—
RQQCF	$10^{-0.1}$	—	30	—	$10^{-0.1}$	—	8	—
SSQSDF	10^{-2}	—	30	—	$10^{-0.3}$	—	30	—
CPCF	10^{-2}	—	30	0.01	$10^{-0.3}$	—	30	0.01
UPCF	10^{-2}	—	30	0.01	$10^{-0.3}$	—	30	0.01

10.1 Experimental Results with Four-Filter Banks

We show filter behavior trends using box-and-whisker plots of the filter responses. Box-and-whisker plots for the four-filter banks over the $[120, 240]$ -deg range are shown in Figs. 4 and 5. These plots show the median (horizontal line in each box), upper and lower quartiles (extents of each box), and upper and lower boundaries (lines extending from the box) of each set of responses as well as the 95% confidence intervals (notches on each box) on the median values. Outliers (values outside the maximum whisker length, equal to 1.5 times the interquartile range) are denoted by \circ (true-class) and \times (false-class). The numbers in parentheses next to the QCF filter type labels indicate the number of eigenvectors N_e used in the filter.

It should be noted that, because these comparisons represent the best possible filter for each design according to the results of Sec. 9.3, they are not computationally fair. For example, the QCFs with $N_e=30$ require 31 FFTs/filter, the PCFs with $q=2$ require 3 FFTs/filter, and the linear filters require 2 FFTs/filter. We carry out equal-computation comparisons in later experiments (see Sec. 10.2 below). While it is difficult to draw conclusions from these box-and-whisker plots, it is nonetheless interesting to observe and visually compare the PSR behavior of the different filters on a common dataset.

Fisher ratios for the $[120, 240]$ -deg range using different sizes of the PSR inner window are shown in Fig. 6. By inspection of these results, we observe that using an inner window of size 7×7 generally provides the best discrimination, although not in every case. For the remainder of our experiments, we use a 7×7 inner window for PSR compu-

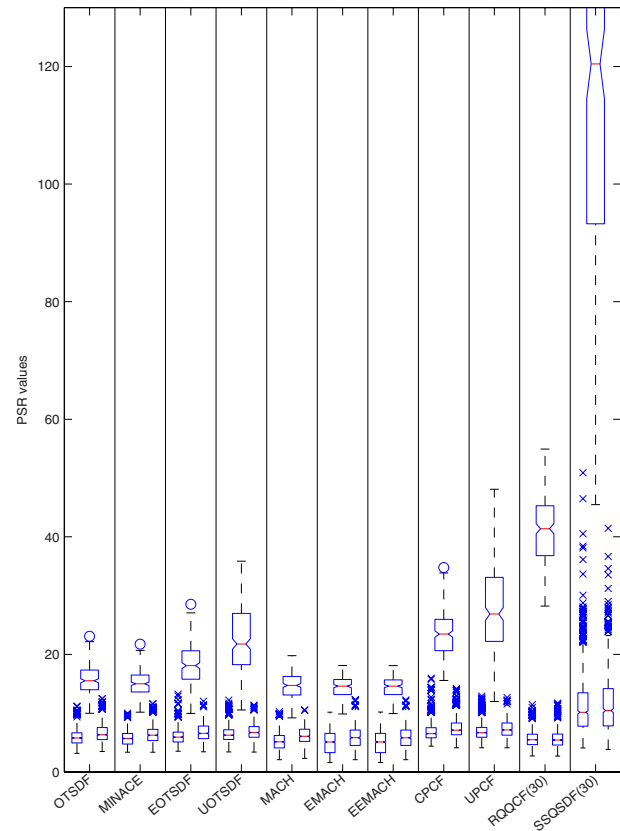


Fig. 4 Box-and-whisker PSR comparison with no noise. The leftmost and rightmost plots in each column correspond to the false-class Leopard and Abrams targets, respectively, while the center plot corresponds to the true-class Missile target.

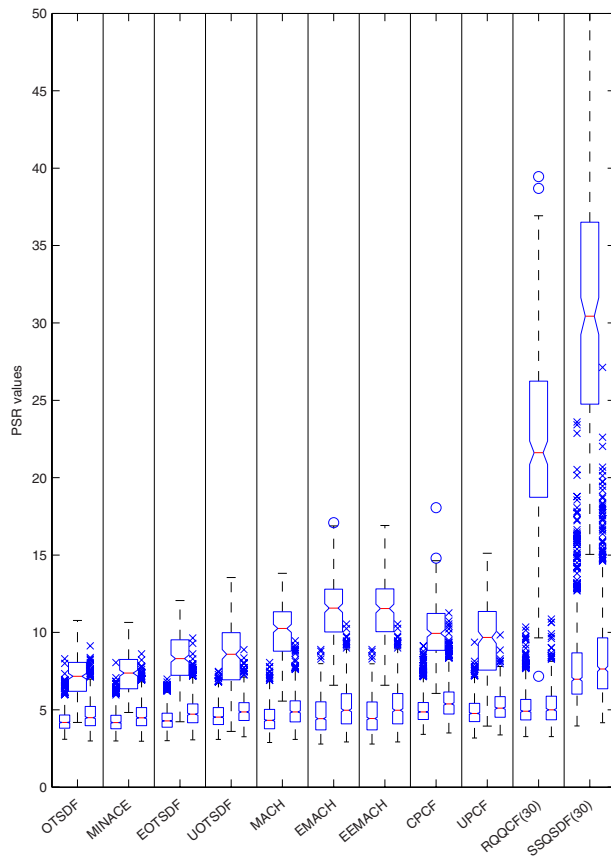


Fig. 5 Box-and-whisker PSR comparison with AWGN at 20-dB SNR. The leftmost and rightmost plots in each column correspond to the false-class Leopard and Abrams targets, respectively, while the center plot corresponds to the true-class Missile target.

tation. As before, the numbers in parentheses next to the QCF filter type labels indicate the number of eigenvectors N_e used in each filter in the bank.

10.2 Experimental Results with 360-deg Banks

New filter banks were trained to divide up the entire 360-deg range of azimuthal rotation based on a given number of filters. For example, if a bank of eight filters were used, the corresponding azimuth ranges for each filter would be $[0, 45], [45, 90], \dots, [315, 360]$ deg. In these experiments, we chose bank sizes and N_e values based on computational restrictions, as explained below. The filters were designed using the same method and parameters as described in Sec. 9.3 (with the exception of the azimuth ranges).

Figure 7 shows the performance of various filter designs at equal computational levels with respect to the number of FFTs required in the use stage. For example, if we limit the number of FFTs per frame to 42 and consider PSR computation, we could use a bank of seven linear filters, six PCFs, or anywhere from one to seven QCFs with varying N_e . We show results for FFT limits of 30, 42, and 72. The number preceding each filter type label in Fig. 7 indicates the number of filters in the bank. In order to facilitate a side-by-side comparison of several RQQCF configurations along with the other designs, we selected only the best design from

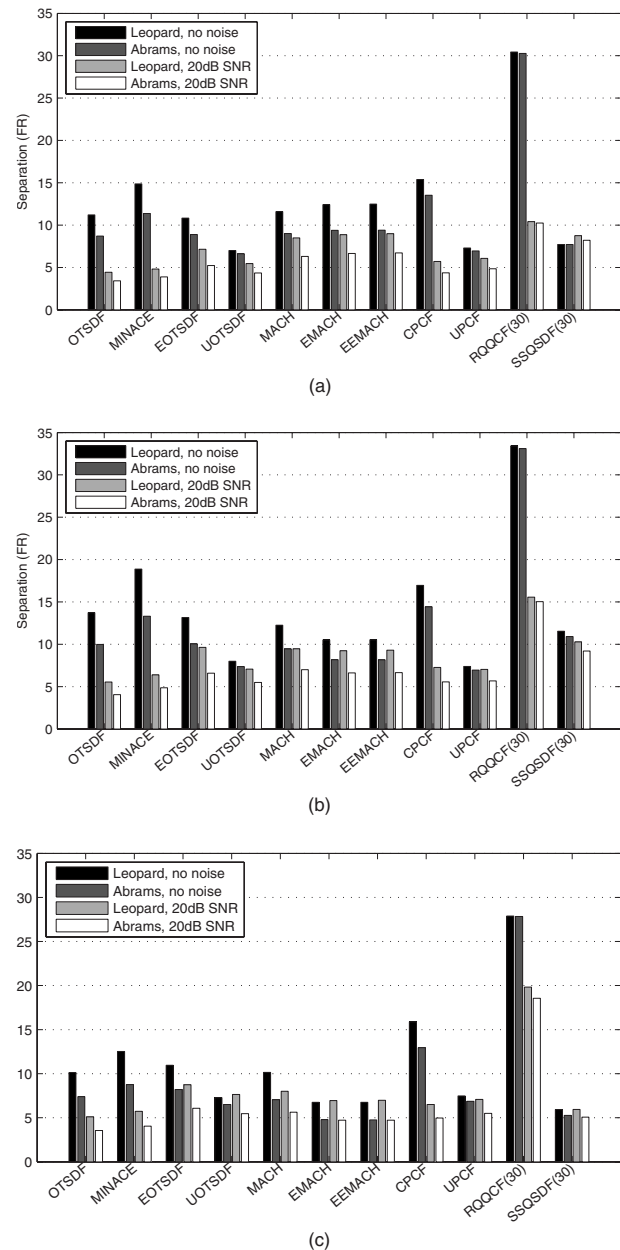


Fig. 6 Fisher separation for each filter using three different sizes for the PSR inner window.

each class (i.e., constrained, unconstrained, eigenfilters, PCF, and QCF) for this final-round comparison.

10.3 Analysis of Filter Comparison

By inspection of the equal-computation performance plots in Fig. 7(c), the RQQCF appears to be the best performer for the rotational data, and the MACH and eigenfilters come in as close seconds to the RQQCF. We make these judgments based on a combination of three factors: (1) performance degradation from clean to noisy data, (2) performance in AWGN, and (3) performance stability across computational levels. We observe that all filters appear to generalize well from seen to unseen false targets. The MACH, EOTSDF, and RQQCF filters exhibit superior noise tolerance to the MINACE and CPCF filters across all

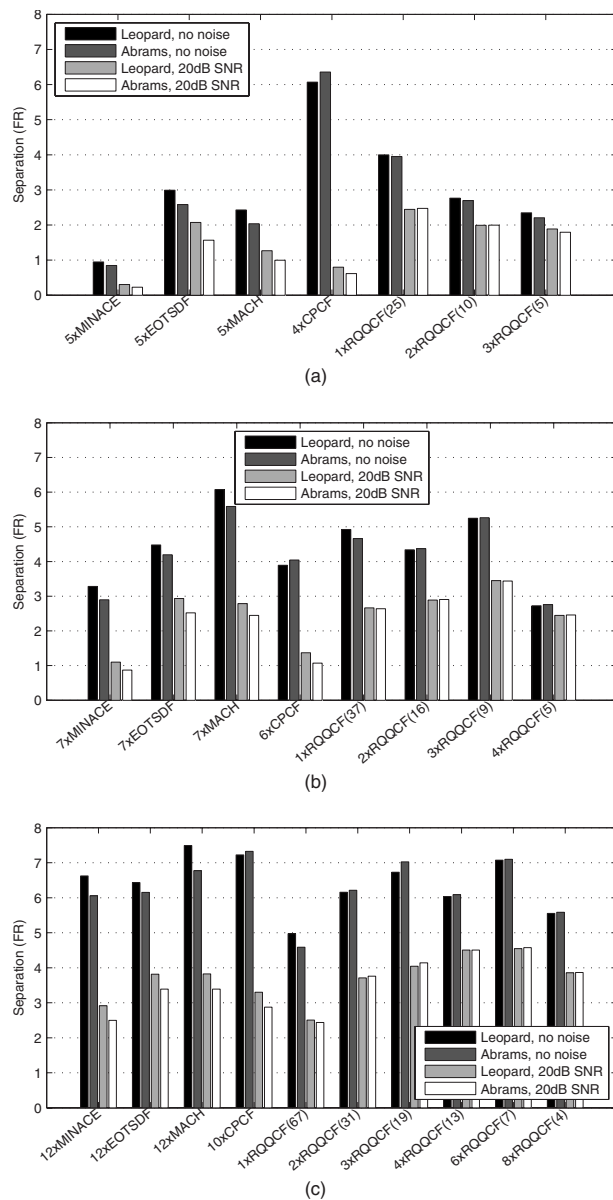


Fig. 7 Equal-computation filter comparison: Fisher separation for selected filter designs over the full 360-deg rotation range using different values for total number of FFTs allowed.

computation levels, suggesting that relaxation of hard constraints in filter design may indeed be advantageous for generalization to noise.

The performance difference between the filters is more stark in the lower-computation scenarios. While the RQCF filters perform better than the other filters in 20-dB AWGN, this advantage decreases as more FFTs are allowed. The constrained filters (CPCF and MINACE) exhibit poor noise tolerance at lower computation levels but perform comparably to the QCFs at higher computation levels. At the 30-FFT level, the constrained filters (7 per bank) are each trained to handle an azimuthal range of 72 deg, while at the 72-FFT level (12 filters per bank), these ranges are each decreased to 30 deg. Thus, the performance results suggest that, with respect to this database, a 72-deg azimuthal range coupled with the included scale

range exceeds the capacity of MINACE and CPCF filters for noisy recognition, whereas a range of 30 deg is within their capacity.

The QCF results in Fig. 7(b) and 7(c) suggest the existence of an optimal trade-off in RQCF design between number of QCFs and number of eigenvectors per filter when computation is limited. For example, a single QCF with a large number of eigenvectors cannot handle the full rotation range as well as two QCFs with fewer (but sufficiently many) eigenvectors each, while on the other hand a very large bank of QCFs with only one or two eigenvectors each will not handle the range well, since the individual QCFs will be poorly approximated.

11 Conclusions

In this paper we have provided a theoretical survey of several composite correlation filter designs. The survey takes a class-based view of composite filters to aid in contrasting the various design methodologies. We supplemented this survey with empirical comparisons of the filter designs on a common data set. This data set contained segmented images of three different scale model targets viewed at different azimuth and depression angles. Our experiments were designed to compare the capacity of each filter design to tolerate out-of-plane rotation distortion.

The combined theoretical survey and empirical results provide an initial basis for selecting an appropriate correlation filter design for a particular application. For example, one application might be limited by computational resources (e.g., recognition on a mobile device), while another had no such limitations. In the former case, eigenfilters might be a better choice of filter, while in the latter case, the higher-computation RQCFs might deliver better results. Alternatively, different applications may exhibit varying levels of noise. While we have only used one specific test set to make our observations, we would expect to see similar results for image data exhibiting similar space-bandwidth product (an information measure) and similar variations in appearance due to such distortions as scale change and rotation.

The main findings regarding filter design and performance are the following:

- Under computationally fair conditions and with respect to Fisher score separation, the RQCFs performed the best among our considered designs on the rotational data set (Sec. 10). They exhibited superior noise performance to the other designs in almost all computationally equal configurations (i.e., N_e versus number of filters). The eigenfilters (EOTSDF) and MACH filters exhibited comparable performance to the RQCF in our experiments. The constrained designs appeared to suffer most from noise in the task of generalization, i.e., a more drastic decrease in generalization performance resulted from adding noise to the test images. These results suggest that relaxing the peak constraints in filter design (e.g., QCF, eigenfilters, unconstrained) may benefit the noise tolerance and generalization ability of the filters without a significant loss of discrimination.
- The performance of the RQCF filter banks appears to be dependent on the number of eigenvectors in-

cluded in the approximation. In other experiments,³⁸ we have observed the existence of an optimum N_e value, beyond which adding eigenvectors leads to decreased performance in noise. This phenomenon was attributed to the fact that later eigenvectors (i.e., those with smaller eigenvalues) tend to have more content in the higher frequencies and are thus more sensitive to noise. We observe that this may lead to an optimal “division of labor” for the RQCFs (Fig. 7), or more specifically, an optimal trade-off between the number of eigenvectors and the size of the distortion range handled by each filter.

An interesting extension of this work would be to include other types of correlation filters (e.g., those that assume a target silhouette with random gray levels) in an empirical comparison. One challenge that would arise is the acquisition of training data that is compatible with all filter designs. Also of interest would be to derive theoretical foundations for predicting the performance of particular filter designs independently of the data, so that empirical evaluations on a particular data set would be unnecessary.

References

1. B. V. K. Vijaya Kumar and A. Mahalanobis, “Recent advances in composite correlation filter design,” *Asian J. Phys.* **8**(3), 407–420 (1999).
2. J. Thornton, M. Savvides, and B. V. K. Vijaya Kumar, “A Bayesian approach to deformed pattern matching of iris images,” *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 596–606 (2007).
3. K. Venkataramani and B. V. K. Vijaya Kumar, “Performance of composite correlation filters for fingerprint verification,” *Opt. Eng.* **43**, 1820–1827 (2004).
4. M. Savvides, J. Heo, J. Thornton, P. Hennings, C. Xie, K. Venkataramani, R. Kerekes, M. Beattie, and B. V. K. Vijaya Kumar, “Biometric identification using advanced correlation filter methods,” in *Springer-Verlag Lecture Notes in Computer Science, Ambient Intelligence* (2005).
5. S. R. F. Sims and A. Mahalanobis, “Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery,” *Opt. Eng.* **43**, 1705–1711 (2004).
6. C. F. Hester and D. Casasent, “Multivariate technique for multiclass pattern recognition,” *Appl. Opt.* **19**, 1758–1761 (1980).
7. A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Casasent, “Minimum average correlation energy filters,” *Appl. Opt.* **26**, 3633–3640 (1987).
8. B. V. K. Vijaya Kumar, “Minimum-variance synthetic discrimination functions,” *J. Opt. Soc. Am. A* **3**(10), 1579–2584 (1986).
9. P. Réfrégier, “Filter design for optical pattern recognition: Multicriteria optimization approach,” *Opt. Lett.* **15**(15), 854–856 (1991).
10. G. Ravichandran and D. Casasent, “Minimum noise and correlation energy (MINACE) optical correlation filter,” *Appl. Opt.* **31**, 1823–1833 (1992).
11. A. Mahalanobis, B. V. K. Vijaya Kumar, S. Song, S. R. F. Sims, and J. F. Epperson, “Unconstrained correlation filters,” *Appl. Opt.* **33**(17), 3751–3756 (1994).
12. M. Alkanhal, B. V. K. Vijaya Kumar, and A. Mahalanobis, “Improving the false alarm capabilities of the maximum average correlation height correlation filter,” *Opt. Eng.* **39**(5), 1133–1141 (2000).
13. R. Shenoy, D. Casasent, and E. G. Zelnio, “Eigen-MINACE SAR detection filters with improved capacity,” *Proc. SPIE* **3370**, 435–447 (1998).
14. A. Mahalanobis, R. Muise, S. R. Stanfill, and A. V. Nevel, “Design and application of quadratic correlation filters for target detection,” *IEEE Trans. Aerosp. Electron. Syst.* **40**(3), 837–850 (2004).
15. R. Shenoy, “The design and use of unconstrained image filters and factures for SAR detection and recognition,” Ph.D. Thesis, Carnegie Mellon Univ., Pittsburgh (2001).
16. B. V. K. Vijaya Kumar, D. W. Carlson, and A. Mahalanobis, “Optimal trade-off synthetic discriminant function filters for arbitrary devices,” *Opt. Lett.* **19**(19), 1556–1558 (1994).
17. B. V. K. Vijaya Kumar, A. Mahalanobis, and A. Takessian, “Optimal tradeoff circular harmonic function correlation filter methods providing controlled in-plane rotation response,” *IEEE Trans. Image Process.* **9**(6), 1025–1034 (2000).
18. B. V. K. Vijaya Kumar and M. Alkanhal, “Eigen-extended maximum average correlation height (EEMACH) filters for automatic target recognition,” *Proc. SPIE* **4379**, 424–431 (2001).
19. R. Muise, A. Mahalanobis, R. Mohapatra, X. Li, D. Han, and W. Mikhael, “Constrained quadratic correlation filters for target detection,” *Appl. Opt.* **43**(2), 304–314 (2004).
20. A. Mahalanobis and B. V. K. Vijaya Kumar, “polynomial filters for higher-order and multi-input information fusion,” in *11th Euro-Am. Opto-Electronic Information Processing Workshop*, pp. 221–231 (1999).
21. K. Al-Mashouq, B. V. K. Vijaya Kumar, and M. Alkanhal, “Analysis of signal-to-noise ratio of polynomial correlation filters,” *Proc. SPIE* **3715**, 407–413 (1999).
22. G. F. Schils and D. W. Sweeney, “Optical processor for recognition of three-dimensional targets viewed from any direction,” *J. Opt. Soc. Am. A* **5**(8), 1309–1321 (1988).
23. G. F. Schils and D. W. Sweeney, “Iterative technique for the synthesis of optical correlation filters,” *J. Opt. Soc. Am. A* **3**, 1433–1442 (1986).
24. L. Hassebrook, B. V. K. Vijaya Kumar, and L. Hostetler, “Linear phase coefficient composite filter banks for distortion-invariant optical pattern recognition,” *Opt. Eng.* **29**(9), 1033–1043 (1990).
25. B. V. K. Vijaya Kumar, D. Casasent, and A. Mahalanobis, “Distance-classifier correlation filters for multiclass target recognition,” *Appl. Opt.* **35**, 3127–3133 (1996).
26. M. Alkanhal and B. V. K. Vijaya Kumar, “Polynomial distance classifier correlation filter for pattern recognition,” *Appl. Opt.* **42**(23), 4688–4708 (2003).
27. R. Singh and B. V. K. Vijaya Kumar, “Performance of the extended maximum average correlation height (EMACH) filter and the polynomial distance classifier correlation filter (PDCCF) for multiclass SAR detection and classification,” in *Algorithms for Synthetic Aperture Radar Imagery IX*, E. G. Zelnio, Ed., *Proc. SPIE* **4727**, 265–276 (2002).
28. F. Goudail and P. Réfrégier, “Optimal detection of a target with random gray levels on a spatially disjoint background noise,” *Opt. Lett.* **21**(7), 495–497 (1996).
29. Y. K. Seong and T. S. Choi, “Optimal-trade-off filters for noise robustness, peak sharpness, and light efficiency in nonoverlapping background noise,” *Opt. Eng.* **39**(2), 472–477 (2000).
30. M. Savvides and B. V. K. Vijaya Kumar, “Efficient design of advanced correlation filters for robust distortion-tolerant face recognition,” in *IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 45–52 (2003).
31. B. V. K. Vijaya Kumar and L. Hassebrook, “Performance measures for correlation filters,” *Appl. Opt.* **29**, 2997–3006 (1990).
32. Demaco, Inc., *Users Manual for Xpatch* (1995).
33. T. D. Ross, S. W. Worrell, V. J. Velt, J. C. Mossing, and M. L. Bryant, “Standard SAR ATR evaluation experiments using the MSTAR public release data set,” *Proc. SPIE* **3370**, 566–573 (1998).
34. E. Tajahuerce, A. Moya, J. Garcia, and C. Ferreira, “Real filter based on Mellin radial harmonics for scale-invariant pattern recognition,” *Appl. Opt.* **33**(14), 3086–3093 (1994).
35. G. Ravichandran and D. Casasent, “Advanced in-plane rotation-invariant correlation filters,” *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(4), 415–420 (1994).
36. R. Kerekes and B. V. K. Vijaya Kumar, “Correlation filters with controlled scale response,” *IEEE Trans. Image Process.* **15**(7), 1794–1802 (2006).
37. R. Hilai and J. Rubinstein, “Recognition of rotated images by invariant Karhunen-Loeve expansion,” *J. Opt. Soc. Am. A* **11**(5), 1610–1618 (1994).
38. R. Kerekes, “Combining correlation outputs for enhanced distortion-tolerant target detection,” Ph.D. Thesis, Carnegie Mellon Univ., Pittsburgh (2007).
39. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley (2001).
40. D. Casasent and X.-W. Chen, “Feature reduction and morphological processing for hyperspectral image data,” *Appl. Opt.* **43**(13), 227–236 (2004).
41. J. Thornton, “Iris matching under deformation and occlusion,” PhD Thesis, Carnegie Mellon Univ., Pittsburgh (2007).
42. The Mathworks, Inc., “MATLAB v7.2,” Software (2006).
43. A. Van Nevel and A. Mahalanobis, “Comparative study of maximum average correlation height filter variants using ladar imagery,” *Opt. Eng.* **42**(2), 541550 (2004).
44. R. Singh, “Advanced correlation filters for multi-class synthetic aperture radar detection and classification,” Master’s Thesis, Carnegie Mellon Univ., Pittsburgh (2002).
45. R. Patnaik and D. Casasent, “MINACE filter classification algorithms for ATR using MSTAR data,” *Proc. SPIE* **5807**, 100–111 (2005).

Ryan A. Kerekes is a postdoctoral research associate in the Computer Sciences and Engineering Division (CSED) at ORNL. He received the PhD degree in electrical and computer engineering from Carnegie Mellon University in 2007, where he studied correlation-based automatic target recognition and tracking. While attending the University of Tennessee, Knoxville, for his undergraduate studies, he worked as an intern in the Image Science and Machine Vision group at ORNL studying small-animal CT imaging. His research interests are in image analysis, computer vision, and machine learning.

B. V. K. Vijaya Kumar received his B.Tech and M.Tech degrees in electrical engineering from Indian Institute of Technology, Kanpur and his Ph.D. in Electrical Engineering from Carnegie Mellon University, Pittsburgh. Since 1982, he has been a faculty member in the Department of Electrical and Computer Engineering (ECE) at Carnegie Mellon where he is now a Professor. Professor Kumar served as the Associate Department Head of the ECE Department from 1994 to 1996 and as the Acting Head of the Department during 2004–2005. Professor Kumar's research interest include Automatic Target Recognition Algorithms, Biometric Recognition Methods and Coding and Signal Processing for Data Storage Systems. He is currently leading coding and signal processing research efforts in the Data Storage Systems Center (DSSC) and the biometrics research efforts in CyLab at Carnegie Mellon. His publications include the book entitled *Correlation Pattern Recognition* (co-authored with Dr. Abhijit Mahalanobis and Dr. Richard Juday, Cambridge University Press, November 2005), twelve book chapters and more than 440 technical papers. He served as a Pattern Recognition Topical Editor for the Information Processing division of *Applied Optics*. He guest-edited the September 1990 special issue of *Optical Engineering* devoted to *Optical Pattern Recognition*, the April 2001 issue of *IEEE Journal of Selected Areas of Communication (JSAC)* devoted to *Signal Processing for High-Density Storage Channels* and the January 2004 special issue of *IEEE Signal Processing* magazine devoted to *iterative soft processing*. He is currently serving as an Associate Editor for *IEEE Transactions on Information Forensics and Security*. Professor Kumar has served on many conference program committees and was co-general chair of the 2004 Optical Data Storage conference, a co-general chair of the 2005 *IEE AutoID Workshop* and is co-chair of the 2008 SPIE conference on *Biometric Technology for Human Identification*. Professor Kumar is a senior member of IEEE, a Fellow of SPIE – The International Society of Optical Engineering, a Fellow of optical Society of America (OSA) and a Fellow of the International Association of Pattern Recognition (IAPR). He is listed in various professional Who's Whos including Marquis' *Who's Who is Science and Engineering* and *Who's Who is the World* and in the *American Men and Women of Sciences*. In 2003, Prof. Kumar received the Eta Kappa Nu award for *Excellence in Teaching* in the ECE Department at CMU and the Carnegie Institute of Technology's *Dowd Fellowship* for educational contributions.