Zero-Aliasing Correlation Filters for Object Recognition

Joseph A. Fernandez, *Student Member, IEEE,* Vishnu Naresh Boddeti, *Member, IEEE*, Andres Rodriguez, *Member, IEEE*, B. V. K. Vijaya Kumar, *Fellow, IEEE*

Abstract—Correlation filters (CFs) are a class of classifiers that are attractive for object localization and tracking applications. Traditionally, CFs have been designed in the frequency domain using the discrete Fourier transform (DFT), where correlation is efficiently implemented. However, existing CF designs do not account for the fact that the multiplication of two DFTs in the frequency domain corresponds to a circular correlation in the time/spatial domain. Because this was previously unaccounted for, prior CF designs are not truly optimal, as their optimization criteria do not accurately quantify their optimization intention. In this paper, we introduce new zero-aliasing constraints that completely eliminate this aliasing problem by ensuring that the optimization criterion for a given CF corresponds to a linear correlation rather than a circular correlation. This means that previous CF designs can be significantly improved by this reformulation. We demonstrate the benefits of this new CF design approach with several important CFs. We present experimental results on diverse data sets and present solutions to the computational challenges associated with computing these CFs. Code for the CFs described in this paper and their respective zero-aliasing versions is available at http://vishnu.boddeti.net/projects/correlation-filters.html

Index Terms—Correlation Filters, Object Recognition, Object Detection, Object Localization.

✦

1 INTRODUCTION

CORRELATION filters (CFs) are a useful tool for a variety of tasks in signal processing and pattern recognition, such as biometric recognition [1], [2], object alignment [3], action recognition [4], object detection [5], [6], [7], object tracking [8], [9] and event retrieval from videos [10]. CFs possess many attractive properties that make them useful for these tasks. Most importantly, CFs are capable of performing both classification and localization simultaneously. In other words, CFs do not assume prior segmentation of objects in test scenes, and are capable of detecting and classifying multiple objects in a single scene simultaneously. CFs are shift-invariant, which means that the objects to be recognized do not have to be centered in the test scene as CFs produce correlation peaks at locations corresponding to the object location in the test scene. These properties make CFs attractive for different applications.

There are two stages in the use of CFs. First, a CF template (a template refers to a 2D array in spatial domain) is designed from a set of training images (we will refer to images and their 2-D Fourier transforms in this paper, but the concepts and methods are equally valid for 1-D temporal signals and higher-dimensional signals such as image sequences). This template is designed such that cross-correlating it with centered positive training images leads to correlation outputs with sharp peaks at the origin and cross-correlating it with negative training

images leads to, either no such discernible peaks or negative peaks. Once designed, the CF is then applied to the test scene, which may be larger than the template itself. The output correlation plane is then searched for positive peaks to determine if any objects from the positive class are detected within the test image. The location of the correlation peak denotes the location of the target; the sharpness of the peak, i.e., how large the peak value is compared to surrounding values, may serve as a measure of confidence regarding the existence of the target.

CFs have traditionally been designed in the spatial frequency domain for computational efficiency. Typically, various metrics are optimized during filter design. One example is minimizing the mean squared error (MSE) between a desired correlation output and the actual correlation output. In formulating these optimizations, cross-correlation in spatial domain is expressed in the frequency domain as the element-wise multiplication of the CF and the conjugate of the Discrete Fourier Transform (DFT) of the training image. Cross-correlating two images of size $N \times M$, using fast Fourier transform (FFT) algorithms (for obtaining 2-D DFTs) requires on the order of $NM \log(NM)$ multiplications whereas a direct spatial-domain correlation requires on the order of $N^2 M^2$ multiplications. However, this DFT-based correlation operation produces a circular correlation rather than a linear correlation [11]. The difference between a linear correlation and a circular correlation can be significant (see Fig. 1). Circular correlation results in aliasing, i.e., parts of linear correlation being added to other parts of itself. This aliasing directly influences the localization performance of the CF. When the CF is *applied* to a test image, aliasing from circular correlation in that operation can be avoided by appropriately padding the template and test image with zeros prior to computing the DFTs. However,

J. A. Fernandez and B. V. K. Vijaya Kumar are with the Department of Electrical and Computer Engineering, and V. N. Boddeti is with the Robotics Institute, Carnegie Mellon University (e-mail: jafernan@andrew.cmu.edu, kumar@ece.cmu.edu, naresh@cmu.edu).

A. Rodriguez is with Air Force Research Laboratory, Dayton, Ohio (e-mail: andres.rodriguez.8@us.af.mil).

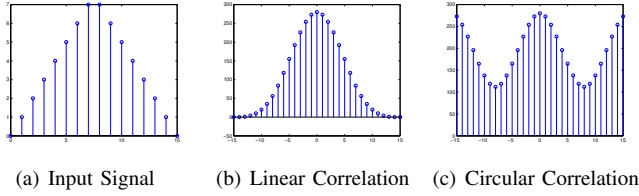| (a) Input Signal | (b) Linear Correlation | (c) Circular Correlation |

Fig. 1. An example illustrating the differences between linear and circular correlation. Here, the input signal in (a) is correlated with itself by appropriately padding the input signal with zeros. The result is the linear correlation output (shown in (b)). This is accomplished with a DFT of size $N = 2 \times 16 - 1 = 31$. The circular correlation output (in part (c)) is generated with a DFT of size 16. Because the signals are not zero-padded, the output in part (c) is a circular correlation. In this example, the effects of aliasing due to circular correlation are very obvious; in other examples, the effects may be less noticeable, but still can affect performance.

simply zero-padding the images does not resolve the aliasing problem in the CF *design* stage. In the past, this fact was mostly ignored, and the optimization needed for CF design was carried out with the incorrect assumption (as we will show in this work) that the circular correlation in CF designs was roughly the same as the linear correlation.

This inconsistency between the CF design and CF use exists not only in the well-known Minimum Average Correlation Energy (MACE) filter [12], but also in many other CF designs, including (but not limited to) the designs presented in [5], [9], [13], [14], [15], [3], [6], [7], and [16]. As we will show in this paper, significantly improved object classification and localization results can be obtained in all CF approaches by removing the aliasing effects due to circular correlation. The problem of circular correlation in CF designs has been mostly ignored for many years; in this paper, we present a solution to this problem. This solution is a *fundamental* and *significant* advancement to the design of CFs. Specifically, we introduce what we term *zero-aliasing (ZA) constraints*, which force the tail of the CF template to zero, which means that the corresponding optimization metric will correspond to a linear correlation rather than a circular correlation. These ZA constraints yield zero-aliasing correlation filters (ZACFs) that are consistent with the original design criteria of existing filter designs. This means that the correlation outputs resemble their intended design. As a result, the performance of ZACFs is significantly improved over conventional CFs. While CFs may not be the best solution for all object recognition problems (e.g., in some problems, the test objects may be centered, thus not benefiting from the CF's localization capabilities), our ZACF approach does offer improvements over existing CF designs that span over multiple decades.

A high level overview of our approach is shown in Fig. 2. We introduced ZA constraints for only the MACE filter in [17]; in this paper, we extend our work on this subject. The contributions of this paper are as follows:

- Extension of ZA constraints to major classes of CF designs (for both scalar and vector features representations),
- Reduced-aliasing correlation filters (RACFs), which offer

a computationally more tractable closed-form solution to the problem, while still allowing some aliasing,
- Fast, efficient and scalable iterative proximal gradient descent based approach for numerically solving ZACF designs,
- A computational and performance comparison of the different methods,
- Numerical results on a variety of datasets demonstrating the superior performance of ZACFs.

Although the main focus of our paper is CFs, we note that many other problems (e.g., convolutional sparse coding [18], convolutional neural networks [19]) which are dependent on convolutions or correlations and are solved in the Fourier domain can potentially benefit from our general observations and the addition of ZA constraints to their respective optimization formulations.

This paper is organized as follows. In Section 2, we summarize related literature. We illustrate the details of circular correlation issue using the well-known MACE filter as an example in Section 3. In Section 4, we show how ZA constraints may be added to several other popular CF designs, namely the Optimal Tradeoff Synthetic Discriminant Function (OTSDF) filter [16]; the Minimum Output Sum of Squared Errors (MOSSE) filter [9], [3], and the Maximum-Margin Correlation Filter (MMCF) [5], [7]. We discuss only a representative sample of CFs, but this approach is applicable to many more CF designs. We discuss some methods to efficiently solve for the ZACF designs in Section 5. In Section 6, we present experimental results for a variety of applications, including face recognition, Automatic Target Recognition (ATR), eye localization and object detection. We conclude in Section 7.

2 RELATED WORK

There are a wide variety of applications for CFs in biometrics, object detection, landmark detection, and action recognition. For example, CFs are used for face recognition in [1]. A deformable pattern matching method is coupled with CFs in [2] to recognize human iris images. More recently, ocular recognition has become of greater interest to the biometric community, and CFs have been applied to this problem in [20]. CFs have also been suggested for biometric key-binding in [21]. Phase-only correlation methods have been used to correlate the phase information in two images. Biometric applications of these techniques include iris recognition [22], as well as face, fingerprint, palmprint, and finger-knuckle recognition [23], [24].

CFs have been used for ATR [5] and to track moving objects in video [25], [9], [8]. Pedestrian detection has been demonstrated in [26], [6]. CFs have been also used as local part detectors. For example, [5], [13] uses CFs for eye localization in images of human faces, and [3] uses CFs as the appearance model for landmark localization in object alignment. CFs have also proven useful for detecting and locating text strings for document processing [27]. In [10] CFs have been used to accurately localize video clips in time thereby dramatically improving (computationally as well as performance wise) video search and retrieval from a large video corpus. Recently
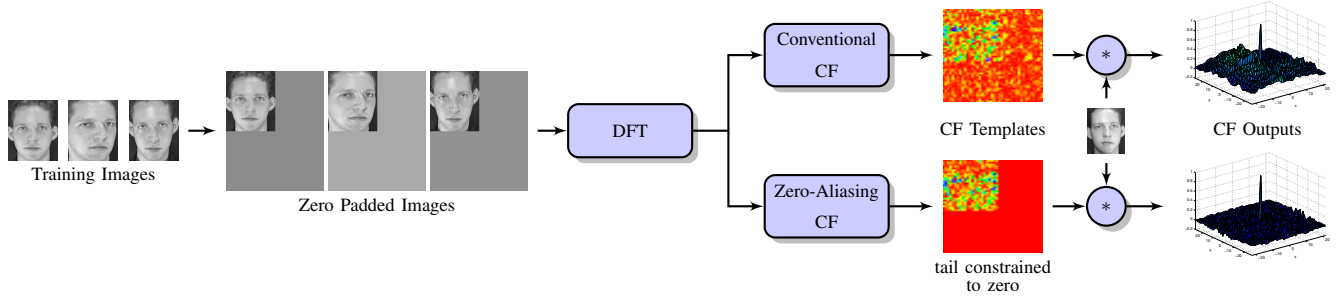
Fig. 2. Overview of our proposed zero-aliasing correlation filters (ZACFs) approach. Conventional CF designs result in templates that are non-zero for all values. This means that, during the optimization, the correlation between the template and the training images is in fact a circular correlation. In our design, constraining the tail of the template using ZA constraints guarantees that the optimization step corresponds to a linear correlation. This results in correlation planes that resemble the original design criteria–in this case, a sharp peak with low side-lobes.

[28] proposed a method to discriminatively learn bases of exemplar correlation filters for accurate 3D pose estimation.

CFs have also been extended to videos and have been applied to detect and classify human actions [4], [29], [30], [31]. In [30], a motion model is combined with two-dimensional CFs to classify human activities. The use of CFs is not limited to pixels; they have been applied to work with feature vectors such as optical flow [4], histogram of oriented gradient (HOG), or scale invariant feature transform (SIFT) features [3], [32].

Despite all the work in CFs, the circular correlation problem has been mostly ignored. This problem was initially observed in [33], which proposed reformulating the popular MACE filter [12] in the spatial domain to avoid circular correlation. However, computation of the MACE filter in the spatial domain is of much higher complexity than computation via the frequency domain. Furthermore, there was never a proposed frequency domain solution to handle the circular correlation problem. More recently, [34] explored several methods to reduce circular correlation effects. These methods use zero padding and/or windowed training images to lessen the effects of circular correlation. One of these methods, originally mentioned in [13], multiplies the training images by a cosine window to reduce edge effects (this approach is explored in greater detail in [34]). Windowing is undesirable, however, because it fundamentally changes the content of the training images. All of the methods discussed in [34] fall short, as they do not eliminate circular correlation effects.

In practice, CFs are typically designed by zero-padding training images (of dimension $N \times M$) to a size of at least $(2N-1)\times(2M-1)$ prior to taking the DFT since zero-padding the input images will result in linear correlation. Although this is true in the testing stage, it is not true in the training stage, and as we will show in Section 3, it does not solve the aliasing problem and results in a suboptimal CF design. In [17], we introduced a new way to completely eliminate aliasing–namely that the CF template must be explicitly constrained during the CF design stage such that the template tail is set to zero. By coupling these ZA constraints with zero-padding the training images, the optimization metric corresponds to a linear correlation, and the resulting CF template is the same as the template we would get if we were to design the CF in the spatial domain. In the next sections, we describe the problem

and our solution in detail.

3 THE CIRCULAR CORRELATION PROBLEM

In this section, we first introduce the main idea that is common to most CF designs and then discuss the circular correlation problem in these designs using the MACE filter [12] as an illustrative example. For notational ease all the expressions through the rest of this paper are given for 1-D signals with K channels. Most prior CF designs were aimed at images (i.e., gray scale values of pixels), but our formulations can accommodate images as well vector features (e.g., HOG, SIFT features) extracted from the images.

Vectors are denoted by lower-case bold ($\mathbf{x}$) and matrices in upper-case bold ($\mathbf{X}$). $\hat{\mathbf{x}} \leftarrow \mathcal{F}_K(\mathbf{x})$ and $\mathbf{x} \leftarrow \mathcal{F}_K^{-1}(\hat{\mathbf{x}})$ denotes the Fourier transform of $\mathbf{x}$ and the inverse Fourier transform of $\hat{\mathbf{x}}$, respectively, where $\hat{\ }$ denotes variables in the frequency domain, $\mathcal{F}_K()$ is the Fourier transform operator and $\mathcal{F}_K^{-1}()$ is the inverse Fourier transform operator with the operators acting on each of the K channels independently. Superscript $\dagger$ denotes the complex conjugate transpose operation.

3.1 Correlation Filter Design

Let us assume that we train a filter using L training images of size N. Typically, we train the filter $\mathbf{h}$ in the frequency domain using DFTs (of size $N_{\mathcal{F}} = N$) of the training images, $\hat{\mathbf{x}}_l$, for $l = 1, \ldots, L$. Many CF designs can be interpreted (see [6] for a circulant decomposition based interpretation and motivation for CF design formulations) as a regression problem optimizing the *localization loss* defined as the MSE between an ideal desired correlation output $\mathbf{g}_l$ for an input image and the correlation output of the training signals with the template[1] (see [3] for details),

$$
\begin{aligned}
J(\mathbf{h}) &= \frac{1}{L} \sum_{l=1}^{L} \left\| \sum_{k=1}^{K} \mathbf{x}_l^k * \mathbf{h}^k - \mathbf{g}_l \right\|_2^2 \quad (1) \\
&= \frac{1}{N_{\mathcal{F}} L} \sum_{l=1}^{L} \left\| \sum_{k=1}^{K} \hat{\mathbf{X}}_l^{k\dagger} \hat{\mathbf{h}}^k - \hat{\mathbf{g}}_l \right\|_2^2 \\
&= \hat{\mathbf{h}}^\dagger \hat{\mathbf{D}} \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger \hat{\mathbf{p}} + E_g
\end{aligned}
$$

1. We refer to the CF as a "template" in the time or spatial domain, and as a "filter" in the frequency domain.

where $*$ denotes the correlation operation, we use the Parseval's theorem to express the MSE in the Fourier domain. E_g is a constant which depends on the desired correlation response, $\hat{\mathbf{X}}_l^{k\dagger}\hat{\mathbf{h}}^k$ is the DFT of the correlation of the k-th channel of the l-th training image with the corresponding k-th channel of the CF template where the diagonal matrix $\hat{\mathbf{X}}_l^k$ contains the vector $\hat{\mathbf{x}}_l^k$ along its diagonal and,

$$\hat{\mathbf{D}} = \frac{1}{N_{\mathcal{F}}L}\begin{bmatrix} \sum_{l=1}^{L}\hat{\mathbf{X}}_l^1\hat{\mathbf{X}}_l^{1\dagger} & \cdots & \sum_{l=1}^{L}\hat{\mathbf{X}}_l^1\hat{\mathbf{X}}_l^{K\dagger} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{L}\hat{\mathbf{X}}_l^K\hat{\mathbf{X}}_l^{1\dagger} & \cdots & \sum_{l=1}^{L}\hat{\mathbf{X}}_l^K\hat{\mathbf{X}}_l^{K\dagger} \end{bmatrix} \quad (2)$$

$$\hat{\mathbf{p}} = \frac{1}{N_{\mathcal{F}}L}\begin{bmatrix} \sum_{l=1}^{L}\hat{\mathbf{X}}_l^1\hat{\mathbf{g}}_l \\ \vdots \\ \sum_{l=1}^{L}\hat{\mathbf{X}}_l^K\hat{\mathbf{g}}_l \end{bmatrix}, \hat{\mathbf{h}} = \begin{bmatrix} \hat{\mathbf{h}}^1 \\ \vdots \\ \hat{\mathbf{h}}^K \end{bmatrix}, \hat{\mathbf{x}}_l = \begin{bmatrix} \hat{\mathbf{x}}_l^1 \\ \vdots \\ \hat{\mathbf{x}}_l^K \end{bmatrix}$$

In addition to minimizing the *localization loss* in Eq. 1, some CFs are constrained to produce pre-specified values at the origin (known as correlation peak constraints) in response to centered training images (this is equivalent to constraining the dot product of the training images and the template).

$$\min_{\hat{\mathbf{h}}} \quad \hat{\mathbf{h}}^\dagger\hat{\mathbf{D}}\hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger\hat{\mathbf{p}} \quad (3)$$
$$s.t. \quad \hat{\mathbf{X}}^\dagger\hat{\mathbf{h}} = \mathbf{u}$$

In this case, $\hat{\mathbf{X}}$ (with no subscripts) is an $N_{\mathcal{F}}K \times L$ matrix with columns $\hat{\mathbf{x}}_l$, and $\mathbf{u}$ is the correlation peak constraint vector, whose elements are usually set to 1 for positive class training images and to 0 for negative class training images (if used). This optimization problem results in a closed form solution for the filter.

$$\hat{\mathbf{h}} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}\left(\hat{\mathbf{X}}^\dagger\hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}\right)^{-1}\mathbf{u} + \quad (4)$$
$$\left(\mathbf{I} - \hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}(\hat{\mathbf{X}}^\dagger\hat{\mathbf{D}}^{-1}\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}^\dagger\right)\hat{\mathbf{D}}^{-1}\hat{\mathbf{p}}$$

The conventional MACE filter [12] is designed to minimize the average correlation energy (ACE, defined below)

$$\text{ACE} = \frac{1}{N_{\mathcal{F}}L}\sum_{l=1}^{L}\sum_{k=1}^{K}\hat{\mathbf{h}}^{k\dagger}\hat{\mathbf{X}}_l^k\hat{\mathbf{X}}_l^{k\dagger}\hat{\mathbf{h}}^k \quad (5)$$

of the entire correlation output (this is equivalent to setting the desired correlation output $\mathbf{g}_l$ to an all-zero plane and equivalently setting $\hat{\mathbf{p}}$ to an all-zero vector) resulting in the following conventional closed form solution for the MACE filter.

$$\hat{\mathbf{h}} = \hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}\left(\hat{\mathbf{X}}^\dagger\hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}\right)^{-1}\mathbf{u} \quad (6)$$

This formulation results in correlation outputs exhibiting sharp correlation peaks with low side-lobes, leading to good localization.

However, there is a fundamental problem with this formulation. The term $\hat{\mathbf{X}}_l^{k\dagger}\hat{\mathbf{h}}^k$ in the ACE expression corresponds to an element-wise multiplication of two DFTs. This term is precisely the problem with the original MACE formulation. When two DFTs are multiplied together in the frequency domain, the result corresponds to a circular correlation in the spatial domain. As shown in Fig.1, circular correlation is an aliased version of the desired linear correlation. To attempt to compensate for the effects of aliasing, the MACE filter is sometimes trained [34] using zero-padded training signals, such that the DFT size is $N_{\mathcal{F}} = (2N - 1)$. Such zero-padding of training signals does not solve the circular correlation problem. This is because, even when training signals are zero-padded, the MACE template obtained using Eq. 6 is usually nonzero for the full template of size $N_{\mathcal{F}}$. Therefore, the multiplication of $\hat{\mathbf{h}}$ and the vectorized DFT of each training signal always results in a circular correlation, regardless of the number of zeros the training signals are padded with. For the term $\hat{\mathbf{X}}_l^{k\dagger}\hat{\mathbf{h}}^k$ to represent a linear correlation $\hat{\mathbf{h}}^k$ must be constrained such that the tail of the spatial domain template $h^k(n)$ is forced to zero. We recently [17] introduced the zero-aliasing MACE (ZAMACE) filter design to eliminate this aliasing problem. The mathematical details are presented in Section 4.2 and extensions to other filter types are presented in Sections 4.3 and 4.4.

3.2 Comparison of MACE and TDMACE Filters

We now explain the intuition and discuss several experiments to illustrate the benefits of our approach using the ZAMACE filter as an example. As a benchmark, we compute the time-domain MACE (TDMACE), which is a template computed in the time domain and is free of any aliasing (for a derivation of this in 1-D, see [17]; a generalized spatial-domain MACE template can be found in [33]; we use the terms "time domain" and "spatial domain" interchangeably). As mentioned previously, computing the MACE filter (or any other CF) in the spatial domain is computationally expensive.

We begin with a 1-D example that uses ECG signals from the MIT-BIH Arrhythmia Database [35]. In this case, we have extracted 355 heartbeat cycles. Each signal is $N = 301$ samples long and represents a single heartbeat, which has been segmented by a cardiologist expert based on the location of the main peak.

We train MACE, TDMACE, and ZAMACE filters using a set of $L = 10$ training signals. Here, we design CFs using training signals from only one class. We desire a correlation template of length N (the same size as the training samples). As mentioned previously, training signals are typically zero-padded to size $N_{\mathcal{F}} \geq 2N - 1$ prior to taking the DFT. However, although the training signals are now zero padded, the conventional MACE filter design does nothing to ensure that the resulting template is zero in its tail. In contrast, our ZA approach (described more formally in Section 4) constrains the optimization of the filter $\hat{\mathbf{h}}$ such that it is zero for the last $N_{\mathcal{F}} - N$ indices of the template $h(n)$. When $N_{\mathcal{F}} \geq 2N - 1$, the circular correlation effects are completely eliminated.

To demonstrate this, we compare the MACE and the ZAMACE formulations while varying q (the amount of zero padding). For the MACE formulation, we demonstrate two methods. First, we train a MACE template of length $N_{\mathcal{F}} = N + q$. Second, we train the same MACE template, but crop it to length N. For the ZAMACE template, we obtain a template of size $N_{\mathcal{F}} = N + q$. However, note that the last q elements of this template are equal to zero because of the ZA

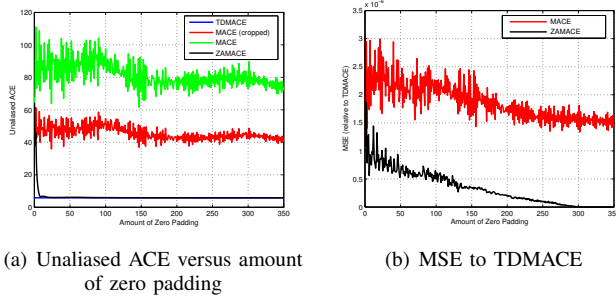(a) Unaliased ACE versus amount of zero padding

(b) MSE to TDMACE

Fig. 3. Comparison of the conventional MACE, TDMACE and ZAMACE filters.

constraints. We correlate each template (in the time domain) with the original training signals and compute the unaliased ACE for each of the formulations. We use the term "unaliased ACE" to make a distinction from ACE, defined in Eq. 5. Unaliased ACE is the average correlation energy of the *linear* correlations of the training samples and the template whereas the ACE term in Eq. 5 is the average correlation energy of the circular correlations. Note that a lower value of unaliased ACE implies sharper correlation peaks, which results in better pattern localization performance.

The results of this analysis are shown in Fig. 3a. First, note that the unaliased ACE for the TDMACE is smaller than both of the conventional frequency-domain MACE formulations for all q values. This indicates that there is a problem with the conventional MACE filter design. Next, note that the unaliased ACE for both of the conventional MACE formulations does not decrease as q increases, which indicates that simply zero padding the training signals does not provide an adequate solution to the aliasing problem. The ZA approach, however, features an unaliased ACE that decreases considerably as q increases. The unaliased ACE of ZAMACE becomes the same as that of the TDMACE when $N_{\mathcal{F}} \geq 2N - 1$, or equivalently when $q \geq 300$. We also show in Fig. 3b the MSE between the (cropped) MACE/ZAMACE templates and the TDMACE template. This cropping is necessary to ensure that the MACE/ZAMACE templates are of the same length as the TDMACE template. Note that the ZAMACE template converges to the TDMACE template as the zero padding approaches 300, whereas the conventional MACE filter differs significantly from TDMACE filter even for $q > 300$. Therefore, the ZAMACE formulation can be interpreted as a bridge between the original MACE formulation and the TDMACE formulation. With no zero padding, there are no ZA constraints and ZAMACE is the same as MACE; with zero-padding $q \geq N-1$, ZAMACE is the same as TDMACE, and there are no circular correlation effects. We repeated this experiment in 2-D, training both MACE and ZAMACE filters with three face images of the same person in the AT&T Database of Faces (formerly the ORL Database of Faces) [36]. We downsampled the images to size 28×23 pixels for computational reasons. We observed the same trend as in the 1-D case–namely, the unaliased ACE decreases as zero padding increases in each dimension. We show the full ($N_{\mathcal{F}} \times M_{\mathcal{F}} = 55 \times 45$) MACE and

ZAMACE templates in Fig. 4. We show example correlation outputs obtained from these MACE and ZAMACE templates in Fig. 5. Note that the correlation output has lower correlation energy for the ZA case than it does for the conventional case. The ZAMACE filter yields an output that is more consistent with the original MACE filter design criteria.

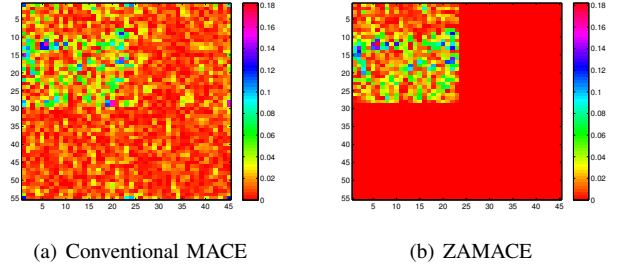(a) Conventional MACE

(b) ZAMACE

Fig. 4. Spatial domain plots of the templates before cropping. Note that the conventional MACE template has large values outside of the ranges of the desired template size, where as the ZAMACE template does not, as the ZA constraints force these values to zero. In these plots, we display the absolute value of the templates.

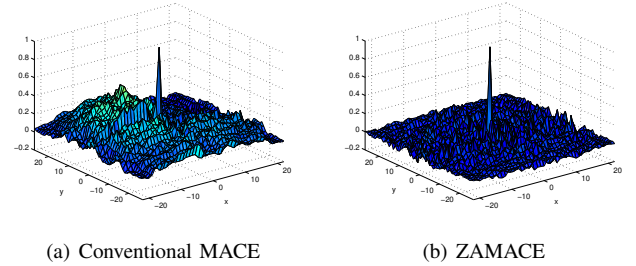

(a) Conventional MACE

(b) ZAMACE

Fig. 5. Example correlation outputs for one of the training images. Note that the ZAMACE filter yields an output that is much sharper than the original MACE filter design.

4 ZERO-ALIASING CORRELATION FILTERS

In this section, we present the mathematical details of ZACFs and demonstrate how ZA constraints can be incorporated into different CF designs. From an optimization perspective we can categorize most CFs into three groups: equality constrained CFs, unconstrained CFs, and inequality constrained CFs. The MACE filter is an example of an equality constrained CF where the dot products between the training images and the template are constrained to equal some pre-specified values. The MOSSE filter [9], [3] is an example of an unconstrained filter that minimizes the *localization loss* between a desired correlation output shape and the actual correlation output. Finally, MMCF [5], [7] is an inequality constrained filter that leverages ideas from support vector machines (SVMs) to achieve better recognition. Although we discuss only a small subset of CFs, similar derivations can be obtained for other CFs.

4.1 Zero-Aliasing Constraints

In the ZACF formulation, we minimize the *localization loss* while enforcing constraints that force the template's tail to be equal to zero. We refer to these constraints as ZA constraints, because they eliminate the aliasing effects caused by circular correlation. For example, for a K-channel template $h(n)$, we can express these constraints as $h^k(n) = 0$ for $n \geq N$ for every k-th channel. Recall that the time domain template $h^k(n)$ is related to the frequency domain filter $H^k(r)$ through the inverse-DFT as,

$$h^k(n) = \frac{1}{N_{\mathcal{F}}} \sum_{r=0}^{N_{\mathcal{F}}-1} H^k(r) e^{\frac{j2\pi rn}{N_{\mathcal{F}}}} \quad (7)$$

To satisfy the ZA constraints, we need to satisfy the following set of linear equations,

$$h^k(n) = \frac{1}{N_{\mathcal{F}}} \sum_{r=0}^{N_{\mathcal{F}}-1} H^k(r) e^{\frac{j2\pi rn}{N_{\mathcal{F}}}} = 0 \text{ for } N \leq n < N_{\mathcal{F}} \quad (8)$$

This can be written in matrix-vector notation as

$$\hat{\mathbf{Z}}^\dagger \hat{\mathbf{h}}^k = \mathbf{0} \quad (9)$$

where

$$\hat{\mathbf{Z}}^\dagger = \begin{bmatrix} 1 & e^{j2\pi(1)(N)/N_{\mathcal{F}}} & \cdots & e^{j2\pi(N_{\mathcal{F}}-1)(N)/N_{\mathcal{F}}} \\ 1 & e^{j2\pi(1)(N+1)/N_{\mathcal{F}}} & \cdots & e^{j2\pi(N_{\mathcal{F}}-1)(N+1)/N_{\mathcal{F}}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j2\pi(1)(N_{\mathcal{F}}-1)/N_{\mathcal{F}}} & \cdots & e^{j2\pi(N_{\mathcal{F}}-1)(N_{\mathcal{F}}-1)/N_{\mathcal{F}}} \end{bmatrix} \quad (10)$$

Note that the $\hat{\mathbf{Z}}^\dagger$ is a matrix of size $(N_{\mathcal{F}} - N) \times N_{\mathcal{F}}$, and $\mathbf{0}$ is a zero vector of length $N_{\mathcal{F}} - N$. Aggregating the ZA constraints from all the K-channels of the signal we have,

$$\hat{\mathbf{A}}^\dagger \hat{\mathbf{h}} = \mathbf{0} \quad (11)$$

where $\hat{\mathbf{A}}^\dagger = \mathbf{I}_K \otimes \hat{\mathbf{Z}}^\dagger$, the kronecker product between $\hat{\mathbf{Z}}^\dagger$ and an identity matrix $\mathbf{I}_K$. These constraints in Eq. 11 are the ZA constraints that characterize ZACFs.

4.2 ZAMACE

In Section 3.1, we discussed the formulation of the MACE filter. In the ZAMACE formulation, we modify the MACE formulation to include the ZA constraints, i.e., we minimize the *localization loss* while enforcing both the peak constraints in Eq. 4 and ZA constraints in Eq. 11 that force the template's tail to be equal to zero.

$$\min_{\hat{\mathbf{h}}} \quad \hat{\mathbf{h}}^\dagger \hat{\mathbf{D}} \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger \hat{\mathbf{p}} \quad (12)$$
$$s.t. \quad \hat{\mathbf{X}}^\dagger \hat{\mathbf{h}} = \mathbf{u}$$
$$\hat{\mathbf{A}}^\dagger \hat{\mathbf{h}} = \mathbf{0}$$

This optimization problem results in a closed form solution for the new ZAMACE filter given by,

$$\hat{\mathbf{h}} = \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}} \left(\hat{\mathbf{B}}^\dagger \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}} \right)^{-1} \mathbf{k} + \quad (13)$$
$$\left(\mathbf{I} - \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}} (\hat{\mathbf{B}}^\dagger \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}})^{-1} \hat{\mathbf{B}}^\dagger \right) \hat{\mathbf{D}}^{-1} \hat{\mathbf{p}}$$

where

$$\hat{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{A}} \end{bmatrix}, \ \mathbf{k} = \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}$$

The ZAMACE expression $\hat{\mathbf{h}} = \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}} \left(\hat{\mathbf{B}}^\dagger \hat{\mathbf{D}}^{-1} \hat{\mathbf{B}} \right)^{-1} \mathbf{k}$ in [17] is obtained by setting the desired correlation outputs $\mathbf{g}_l$ to all-zero plane which leads to $\hat{\mathbf{p}} = \mathbf{0}$. The size of the filter, $\hat{\mathbf{h}}$, in Eq. 13 is the same as in Eq. 6. This means that the ZAMACE filter requires no additional storage, and for testing it requires no additional computation or memory than the MACE filter.

Note that the OTSDF filter [16] is a CF that is very similar to the MACE filter. In the OTSDF formulation, the matrix $\hat{\mathbf{D}}$ is replaced by $\hat{\mathbf{T}} = \hat{\mathbf{D}} + \delta \mathbf{I}$, where $\mathbf{I}$ is an identity matrix and $\delta > 0$. The inclusion of the identity matrix is to improve noise tolerance.

4.3 ZAMOSSE

The MOSSE filter [9], [3], [32] is an unconstrained filter that minimizes the *localization loss*, defined in Eq. 1, between the correlation of the CF template with the training signal(s) and the desired correlation output(s). For example, the desired correlation output in the spatial domain could take on a value of one at the location of a positive class signal and zeros elsewhere. Similar filters that specify a desired correlation output are found in [13], [37], [15], and [21]. The MOSSE filter design is formulated as the following optimization problem,

$$\min_{\hat{\mathbf{h}}} \quad \hat{\mathbf{h}}^\dagger \hat{\mathbf{D}} \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger \hat{\mathbf{p}} \quad (14)$$

which results in the following closed-form expression for the filter,

$$\hat{\mathbf{h}} = \hat{\mathbf{D}}^{-1} \hat{\mathbf{p}} \quad (15)$$

This expression is written in matrix-vector notation, but is equivalent to the expression in [9] for a single channel signal.

Again, this formulation does not account for the aliasing effects due to circular correlation. To remove the effects of circular correlation, we need to instead optimize Eq. 14 subject to the ZA constraints given in Eq. 11. The ZAMOSSE filter design results in the following closed form expression,

$$\hat{\mathbf{h}} = \mathbf{\Delta}_{\hat{\mathbf{D}}} \hat{\mathbf{D}}^{-1} \hat{\mathbf{p}} \quad (16)$$

where $\mathbf{\Delta}_{\hat{\mathbf{D}}} = \mathbf{I} - \hat{\mathbf{D}}^{-1} \hat{\mathbf{A}} (\hat{\mathbf{A}}^\dagger \hat{\mathbf{D}}^{-1} \hat{\mathbf{A}})^{-1} \hat{\mathbf{A}}^\dagger$. In our MOSSE and ZAMOSSE filter implementation, we replace $\hat{\mathbf{D}}$ with $\hat{\mathbf{T}} = \hat{\mathbf{D}} + \delta \mathbf{I}$ as described earlier.

4.4 ZAMMCF

The Maximum-Margin Correlation Filter was recently introduced in [5], [7]. This filter combines the localization ability of the CFs and the generalization capability of large-margin based classifiers like Support Vector Machines (SVMs). Traditionally, constrained CF designs (e.g., MACE) are constrained such that the dot product of a training image and the CF template is set to a specific value. In the MMCF formulation, however, this hard equality constraint is relaxed and replaced with inequality constraints for maximizing the margin of

separation between the positive and negative class training samples.

$$\min_{\hat{\mathbf{h}}} \quad \hat{\mathbf{h}}^\dagger \hat{\mathbf{T}} \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger \hat{\mathbf{p}} + 2C\mathbf{1}^T \boldsymbol{\xi} \quad (17)$$

$$s.t. \quad y_l \left(\hat{\mathbf{x}}_l^\dagger \hat{\mathbf{h}} + b \right) \geq y_l u_l - \xi_l$$

$$\xi_l \geq 0$$

where $\hat{\mathbf{T}} = \hat{\mathbf{D}} + \delta\mathbf{I}$ as described before, the vector $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_L]^T$ is a real vector of slack variables that penalize training images that are on the wrong side of the margin (as in the SVM formulation), $C > 0$ is a trade-off parameter, y_l is the class label (1 for positive class and -1 for negative class), and u_l is the minimum peak magnitude which is typically set to 1. As shown in [38], the filter may be expressed as

$$\hat{\mathbf{h}} = \hat{\mathbf{T}}^{-1}\hat{\mathbf{p}} + \hat{\mathbf{T}}^{-1}\hat{\mathbf{X}}\mathbf{Y}\mathbf{a} \quad (18)$$

where $\mathbf{Y}$ is a diagonal matrix with the class labels y_l along the diagonal, and $\mathbf{a}$ is the solution of the dual problem,

$$\max_{\mathbf{0} \leq \mathbf{a} \leq C\mathbf{1}} \mathbf{a}^T \mathbf{M} \mathbf{a} + \mathbf{a}^T \mathbf{d} \quad (19)$$

where

$$\mathbf{M} = -\mathbf{Y}\hat{\mathbf{X}}^\dagger \hat{\mathbf{T}}^{-1}\hat{\mathbf{X}}\mathbf{Y}$$

$$\mathbf{d} = 2\mathbf{Y}(\mathbf{u} - \hat{\mathbf{X}}^\dagger \hat{\mathbf{T}}^{-1}\hat{\mathbf{p}})$$

To obtain the ZA formulation of MMCF (ZAMMCF), we add the ZA constraints, Eq. 11, to the MMCF formulation, Eq. 17. The solution to this problem can be expressed as,

$$\hat{\mathbf{h}} = \hat{\mathbf{T}}^{-1}(\hat{\mathbf{p}} + \hat{\mathbf{X}}\mathbf{Y}\mathbf{a} + \hat{\mathbf{A}}\boldsymbol{\omega}) \quad (20)$$

where $\boldsymbol{\omega} = -(\hat{\mathbf{A}}^\dagger \hat{\mathbf{T}}^{-1}\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^\dagger(\hat{\mathbf{T}}^{-1}\hat{\mathbf{p}} + \hat{\mathbf{T}}^{-1}\hat{\mathbf{X}}\mathbf{Y}\mathbf{a})$ and $\mathbf{a}$ is the solution of the dual problem,

$$\max_{\mathbf{0} \leq \mathbf{a} \leq C\mathbf{1}} \mathbf{a}^T \mathbf{M} \mathbf{a} + \mathbf{a}^T \mathbf{d} \quad (21)$$

where

$$\mathbf{M} = -\mathbf{Y}\hat{\mathbf{X}}^\dagger \boldsymbol{\Delta}_{\hat{\mathbf{T}}} \hat{\mathbf{T}}^{-1}\hat{\mathbf{X}}\mathbf{Y}$$

$$\mathbf{d} = 2\mathbf{Y}\left(\mathbf{u} - \hat{\mathbf{X}}^\dagger \boldsymbol{\Delta}_{\hat{\mathbf{T}}} \hat{\mathbf{T}}^{-1}\hat{\mathbf{p}}\right)$$

$$\boldsymbol{\Delta}_{\hat{\mathbf{T}}} = \mathbf{I} - \hat{\mathbf{T}}^{-1}\hat{\mathbf{A}}(\hat{\mathbf{A}}^\dagger \hat{\mathbf{T}}^{-1}\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^\dagger$$

4.5 Extension to Multi-Dimensional Signals

Extending the ZA formulation to multi-dimensional signals is a matter of determining the matrix $\hat{\mathbf{A}}$ such that $\hat{\mathbf{A}}^\dagger \hat{\mathbf{h}} = \mathbf{0}$ when $\hat{\mathbf{h}}$ is a vectorized M-D DFT. We illustrate the 2-D ZA constraints in Fig. 6. Here, the shaded region of the template must be set to zero via ZA constraints. We can express $\mathbf{h}$ (the vectorized template) as

$$\mathbf{h} = \mathbf{W}\hat{\mathbf{h}} \quad (22)$$

where the matrix $\mathbf{W}$ accomplishes a 2-D inverse-DFT. Note that we only wish to constrain some of the entries of vector $\mathbf{h}$ to zero. The rows of matrix $\hat{\mathbf{A}}^\dagger$ are taken from the rows of matrix $\mathbf{W}$ corresponding to the elements of $\mathbf{h}$ that we wish to constrain. Details may be found in [17]. Note that matrix $\hat{\mathbf{A}}$ is considerably larger in 2-D thereby posing a computational challenge for numerically computing ZACFs.
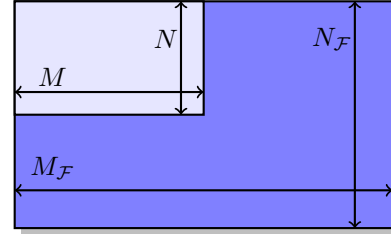

Fig. 6. Illustration of the ZA constraints required for the 2-D formulation of the ZAMACE template.

5 COMPUTATIONAL CONSIDERATIONS

In this section, we discuss the computational challenges that arise in the design of ZACFs. Although we have analytical solutions for ZACFs, designing the filter can become computationally intensive (memory wise), especially for large training images. *Note that applying the resulting ZACF to test data requires no additional computation or memory compared to traditional CFs. The computational considerations presented in this section are only for training the CF.* In this section, we present two different methods we have developed to overcome the computational challenge and efficiently design ZACFs.

5.1 Reduced-Aliasing Correlation Filters

Analytical solutions for ZACFs typically result in greater computational complexity than their conventional counterparts. To illustrate this, we use the MACE filter as an example. In the original MACE formulation, the matrix $\hat{\mathbf{X}}^\dagger \hat{\mathbf{D}}^{-1}\hat{\mathbf{X}}$ must be inverted. This is usually of low computational complexity, as the size of this matrix ($L \times L$) is determined only by the number of training images, L, which is usually much smaller than the number of pixels in the training images. For the ZAMACE case, however, we must invert the matrix $\hat{\mathbf{B}}^\dagger \hat{\mathbf{D}}^{-1}\hat{\mathbf{B}}$, which is of size $(L + KN_{\mathcal{F}} - KN) \times (L + KN_{\mathcal{F}} - KN)$, which is usually much larger than the MACE case. Note that if we reduce the number of ZA constraints ($N_{\mathcal{F}} - N$), we can reduce the size of this matrix. We have explored several ways to do this. First, we explored constraining only a portion of the template tail based on the observation that more energy is usually contained in the main portion of the template. However, this technique is not nearly as successful as simply reducing the DFT size ($N < N_{\mathcal{F}} < 2N - 1$) and constraining the entire tail. Doing so means that there is still aliasing due to circular correlation; however, we have noticed that this aliasing is significantly reduced, leading to very good results while improving the computational requirements. We refer to this method as the reduced-aliasing correlation filter (RACF). RACFs were inspired by our earlier experiments (see Fig. 3a), in which we noticed that the unaliased ACE actually decreases rapidly as zero-padding increases. We evaluate the RACF method in Section 5.3.

5.2 Proximal Gradient Descent Method

Although we have derived closed-form solutions for some ZACFs, sometimes implementing these expressions may be impractical from a computational and memory perspective. An

alternative solution is to use an iterative algorithm to solve for the ZACF. One iterative method to do this is using the proximal gradient descent method [39]. We have developed this method for unconstrained filters (ZAMOSSE), equality constrained filters (ZAMACE/ZAOTSDF) and inequality constrained filters (ZAMMCF). The key idea to efficiently optimize the ZACF formulations is to impose the ZA constraints directly in the spatial domain instead of forming the matrix $\hat{\mathbf{A}}$ as in the closed form solution. Imposing the constraints in the spatial domain can be done by a simple projection of the template onto the space Ω whose tail is set to zero. We define $P_\Omega(\mathbf{x})$ as the operator which projects $\mathbf{x}$ onto the set Ω i.e., $P_\Omega(\mathbf{x})$ is the operator that sets the tail of $\mathbf{x}$ to zero. Further, $P_Z(\mathbf{x})$ and $P_C(\mathbf{x})$ are defined as complementary operators which zero-pad and crop the signal by an appropriate amount respectively.

As described earlier, most CF designs optimize the *localization loss*, $f(\hat{\mathbf{h}}) = \hat{\mathbf{h}}^\dagger \hat{\mathbf{T}} \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^\dagger \hat{\mathbf{p}}$, whose gradient is $\nabla f(\hat{\mathbf{h}}) = 2\hat{\mathbf{T}}\hat{\mathbf{h}} - 2\hat{\mathbf{p}}$. Standard gradient descent [40] finds an optimal solution $\hat{\mathbf{h}}_{opt}$ by choosing an initial solution, $\hat{\mathbf{h}}_0$ and iteratively reducing the cost function $f(\hat{\mathbf{h}})$. Each iteration is computed as a function of the previous solution,

$$\hat{\mathbf{h}}_{t+1} = \hat{\mathbf{h}}_t - \eta_t \nabla f\left(\hat{\mathbf{h}}_t\right) \tag{23}$$

where η_t is the step size at iteration t. However, standard gradient descent does not allow for the ZA constraints to be imposed on the filter. Therefore, we use proximal gradient descent [39] to find the optimal solution while satisfying the ZA constraints,

$$\hat{\mathbf{h}}_{t+1} = prox\left(\hat{\mathbf{h}}_t - \eta_t \nabla f\left(\hat{\mathbf{h}}_t\right)\right) \tag{24}$$

Here, the $prox()$ operator imposes constraints on the filter.

For unconstrained filters (e.g., ZAMOSSE), the $prox()$ operator (see Fig. 7a and Algorithm 1) transforms the filter update $\mathcal{F}^{-1}\left(\hat{\mathbf{h}}_t - \eta_t \nabla f\left(\hat{\mathbf{h}}_t\right)\right) \to \mathbf{h}_{t+\frac{1}{2}}$ into the spatial domain and sets the template's tail to zero i.e., $\mathbf{h}_{t+\frac{1}{2}} \leftarrow P_\Omega\left(\mathbf{h}_{t+\frac{1}{2}}\right)$. It then transforms the resulting template back into the frequency domain to obtain $\hat{\mathbf{h}}_{t+1} \leftarrow \mathcal{F}\left(\mathbf{h}_{t+\frac{1}{2}}\right)$. The full algorithm is described in Algorithm 1.

For the equality constrained case (e.g. ZA-MACE/ZAOTSDF), the $prox()$ operator contains two steps (see Fig. 7b and Algorithm 2). First, the filter update is transformed into the spatial domain $\mathcal{F}^{-1}\left(\hat{\mathbf{h}}_t - \eta_t \nabla f\left(\hat{\mathbf{h}}_t\right)\right) \to \mathbf{h}_{t+\frac{1}{2}}$ and the template's tail is set to zero i.e., $\mathbf{h}_{t+\frac{1}{2}} \leftarrow P_\Omega\left(\mathbf{h}_{t+\frac{1}{2}}\right)$. For the second step, we extract the main portion of the template and vectorize it i.e., $\mathbf{h}^{\#}_{t+\frac{1}{2}} \leftarrow P_C\left(\mathbf{h}_{t+\frac{1}{2}}\right)$. We now seek to update $\mathbf{h}^{\#}_{t+\frac{1}{2}}$ by $\mathbf{h}_\Delta$ such that the desired peak constraints are satisfied and that is closest to the current template $\mathbf{h}^{\#}_t$,

$$\mathbf{X}^T\left(\mathbf{h}^{\#}_{t+\frac{1}{2}} + \mathbf{h}_\Delta\right) = \mathbf{u} \tag{25}$$

where $\mathbf{u}$ is a vector containing the desired peak constraints and $\mathbf{X}$ contains the training signals in the spatial domain along

(a) Unconstrained and Inequality Constrained Case
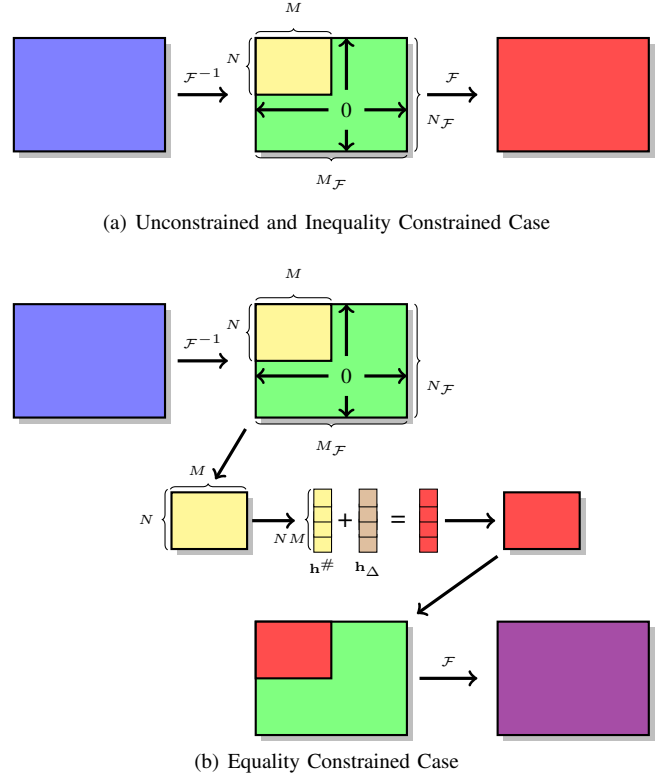

(b) Equality Constrained Case

Fig. 7. Illustration of the proximal step, which is performed in the spatial domain. All DFTs and Inverse-DFTs are in 2-D.

each column. We solve for $\mathbf{h}_\Delta$ by solving the following least squares problem,

$$\min_{\mathbf{h}_\Delta} \quad \mathbf{h}_\Delta^T \mathbf{h}_\Delta \tag{26}$$
$$s.t. \quad \mathbf{X}^T \mathbf{h}_\Delta = \mathbf{u} - \mathbf{X}^T \mathbf{h}^{\#}_{t+\frac{1}{2}}$$

which results in $\mathbf{h}_\Delta = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\left(\mathbf{u} - \mathbf{X}^T\mathbf{h}^{\#}_{t+\frac{1}{2}}\right)$. Next, we form the vector $\mathbf{h}^{\#}_{t+\frac{1}{2}} + \mathbf{h}_\Delta$, zero-pad it by the appropriate amount to get $\mathbf{h}_{t+1} \leftarrow P_Z\left(\mathbf{h}^{\#}_{t+\frac{1}{2}} + \mathbf{h}_\Delta\right)$, and map it back to the Fourier domain $\hat{\mathbf{h}}_{t+1} \leftarrow \mathcal{F}\left(\mathbf{h}_{t+1}\right)$. The full algorithm is described in Algorithm 2.

For the inequality constrained case (e.g., ZAMMCF), since the hinge loss is non-differentiable, gradient descent cannot be directly used. While many sub-gradient descent methods have been developed for the hinge loss, we empirically observed that a simple sub-gradient descent with the proximal operator had poor convergence properties. Therefore, we instead use the differentiable *squared hinge loss* and solve the following optimization problem that is closely related to the formulation in Eq. 27,

$$\min_{\hat{\mathbf{h}}} \quad \frac{\lambda}{2}\hat{\mathbf{h}}^\dagger \hat{\mathbf{T}} \hat{\mathbf{h}} + \frac{1}{2L}\sum_{l=1}^{L}\left[1 - y_l\left(\hat{\mathbf{x}}_l^\dagger \hat{\mathbf{h}} + b\right)\right]_+^2 \tag{27}$$
$$s.t. \quad \hat{\mathbf{A}}^\dagger \hat{\mathbf{h}} = \mathbf{0}$$

where $\lambda = \frac{1}{C}$. Again we adopt accelerated gradient descent along with the proximal step after each iteration to solve for the filter. The gradient of this objective function is,

$$\nabla f(\hat{\mathbf{h}}) = \lambda \hat{\mathbf{T}} \hat{\mathbf{h}} - \frac{1}{L} \sum_{i \in \Omega_{sv}} y_i \hat{\mathbf{x}}_i \left(1 - y_i \left(\hat{\mathbf{h}}^\dagger \hat{\mathbf{x}}_i + b\right)\right) \quad (28)$$

where the support vector set Ω_{sv} is defined as,

$$\Omega_{sv} = \left\{ i \in (1, \dots, n) \; \middle| \; y_i \left(\hat{\mathbf{h}}^\dagger \hat{\mathbf{x}}_i + b\right) < 1 \right\}$$

We estimate the step-size η_t for gradient descent by exact line search i.e.,

$$\min_\eta f\left(\hat{\mathbf{h}}_t + \eta \nabla f\left(\hat{\mathbf{h}}_t\right)\right) \quad (29)$$

which we solve by a 1-D Newton method. The full algorithm is described in Algorithm 3.

Algorithm 1 Accelerated Proximal Gradient Descent for Unconstrained Filters (ZAMOSSE).

Compute $\hat{\mathbf{h}}_{conv}$ (conventional CF solution)
Initialize $\hat{\mathbf{v}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$ and $\hat{\mathbf{w}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$
repeat
    $\nabla f(\hat{\mathbf{w}}_t) = 2\hat{\mathbf{T}}\hat{\mathbf{w}}_t - 2\hat{\mathbf{p}}$
    $\eta_t \leftarrow$ exact line search
    $\hat{\mathbf{v}}_{t+1} = prox\left(\hat{\mathbf{w}}_t - \eta_t \nabla f(\hat{\mathbf{w}}_t)\right)$
    $\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{v}}_{t+1} + \left(\frac{t-1}{t+2}\right)(\hat{\mathbf{v}}_{t+1} - \hat{\mathbf{v}}_t)$
until $\frac{|f(\hat{\mathbf{v}}_{t+1}) - f(\hat{\mathbf{v}}_t)|}{|f(\hat{\mathbf{v}}_t)|} < \epsilon$
Output $\hat{\mathbf{h}} \leftarrow \hat{\mathbf{v}}_{t+1}$

Function $prox(\hat{\mathbf{h}})$
1: $\mathbf{h} \leftarrow \mathcal{F}(\hat{\mathbf{h}})$
2: $\mathbf{h} \leftarrow P_\Omega(\mathbf{h})$
3: $\hat{\mathbf{h}} \leftarrow \mathcal{F}^{-1}(\mathbf{h})$

Algorithm 2 Accelerated Proximal Gradient Descent for Equality Constrained Filters (ZAMACE and ZAOTSDF).

Compute $\hat{\mathbf{h}}_{conv}$ (conventional CF solution)
Initialize $\hat{\mathbf{v}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$ and $\hat{\mathbf{w}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$
repeat
    $\nabla f(\hat{\mathbf{w}}_t) = 2\hat{\mathbf{T}}\hat{\mathbf{w}}_t - 2\hat{\mathbf{p}}$
    $\eta_t \leftarrow$ exact line search
    $\hat{\mathbf{v}}_{t+1} = prox\left(\hat{\mathbf{w}}_t - \eta_t \nabla f(\hat{\mathbf{w}}_t)\right)$
    $\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{v}}_{t+1} + \left(\frac{t-1}{t+2}\right)(\hat{\mathbf{v}}_{t+1} - \hat{\mathbf{v}}_t)$
until $\frac{|f(\hat{\mathbf{v}}_{t+1}) - f(\hat{\mathbf{v}}_t)|}{|f(\hat{\mathbf{v}}_t)|} < \epsilon$
Output $\hat{\mathbf{h}} \leftarrow \hat{\mathbf{v}}_{t+1}$

Function $prox(\hat{\mathbf{h}})$
1: $\mathbf{h} \leftarrow \mathcal{F}(\hat{\mathbf{h}})$
2: $\mathbf{h} \leftarrow P_\Omega(\mathbf{h})$
3: $\mathbf{h}^\# \leftarrow P_C(\mathbf{h})$
4: $\mathbf{h}_\Delta = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{u} - \mathbf{X}^T\mathbf{h}^\#)$
5: $\mathbf{h}^\# \leftarrow \mathbf{h}^\# + \mathbf{h}_\Delta$
6: $\mathbf{h}^\# \leftarrow P_Z(\mathbf{h})$
7: $\hat{\mathbf{h}} \leftarrow \mathcal{F}^{-1}(\mathbf{h})$

Algorithm 3 Accelerated Proximal Gradient Descent for Inequality Constrained Filters (ZAMMCF).

Compute $\hat{\mathbf{h}}_{conv}$ (any conventional CF solution)
Initialize $\hat{\mathbf{v}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$ and $\hat{\mathbf{w}}_0 \leftarrow prox(\hat{\mathbf{h}}_{conv})$
repeat
    $\Omega_{sv} \leftarrow \left\{ i \in (1, \dots, n) \; \middle| \; \left[1 - y_i(\hat{\mathbf{w}}_k^\dagger \hat{\mathbf{x}}_i + b)\right] < 0 \right\}$
    $\nabla f(\hat{\mathbf{w}}_k) = \lambda \hat{\mathbf{T}}\hat{\mathbf{w}}_k - \frac{1}{L} \sum_{i \in \Omega_{sv}} y_i \hat{\mathbf{x}}_i (1 - y_i(\hat{\mathbf{w}}_k^\dagger \hat{\mathbf{x}}_i + b))$
    $\eta_t \leftarrow$ exact line search
    $\hat{\mathbf{v}}_{t+1} = prox\left(\hat{\mathbf{w}}_t - \eta_t \nabla f(\hat{\mathbf{w}}_t)\right)$
    $\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{v}}_{t+1} + \left(\frac{t-1}{t+2}\right)(\hat{\mathbf{v}}_{t+1} - \hat{\mathbf{v}}_t)$
until $\frac{|f(\hat{\mathbf{v}}_{t+1}) - f(\hat{\mathbf{v}}_t)|}{|f(\hat{\mathbf{v}}_t)|} < \epsilon$
Output $\hat{\mathbf{h}} \leftarrow \hat{\mathbf{v}}_{t+1}$

Function $prox(\hat{\mathbf{h}})$
1: $\mathbf{h} \leftarrow \mathcal{F}(\hat{\mathbf{h}})$
2: $\mathbf{h} \leftarrow P_\Omega(\mathbf{h})$
3: $\hat{\mathbf{h}} \leftarrow \mathcal{F}^{-1}(\mathbf{h})$

Our proximal gradient descent approach ensures that the ZA constraints (and peak constraints, for constrained filters) are satisfied. This is done in the spatial domain, rather than forming matrix $\hat{\mathbf{A}}$ (or matrix $\hat{\mathbf{B}}$, for the constrained case) as in the closed form solution. This is advantageous because we save on the memory needed to compute and store $\hat{\mathbf{A}}$ (or $\hat{\mathbf{B}}$) and the subsequent computational resources needed to solve large systems of equations. The result is a memory-stable, low complexity solution that allows for fast, efficient and scalable computation of ZACFs.

5.3 Computational Experiments

To demonstrate the benefits of RACF and the proximal gradient descent methods, we perform a computational analysis in this section (see Section 6 for accuracy comparisons) to compare both methods to the closed-form ZACF implementation. For RACF, we pad with a number of zeros equal to 10% or 25% of the training image size, and refer to these results as RACF-10% and RACF-25%, respectively. For the proximal gradient method, we compute the step size η_t via exact line search. We initialize the filter ($\hat{\mathbf{h}}_0$) as the conventional filter design subjected to the $prox()$ operator. We use a stopping condition $\frac{|f(\hat{\mathbf{v}}_k) - f(\hat{\mathbf{v}}_{k-1})|}{|f(\hat{\mathbf{v}}_{k-1})|} < 10^{-10}$ to terminate the optimization. In this experiment, we used a training size of 9 images from the AT&T (ORL) Face Dataset [36]; we vary the resolution and crop the training images so that they are square. The platform for this experiment was a desktop running Matlab 2011a with Windows 7, an Intel Core i7-2600 CPU (3.4 GHz), and 16 GB of RAM. We show the results of our experiment in Fig. 8. We only compute the closed form solution for a resolution up to 50×50, as this is near the memory limit of our machine. Note that, while the fastest method is RACF-10%, we will show in Section 6 that RACF-25% will perform better from a recognition performance perspective. However, recognition performance is dependent on the image size and the training set, so experimentation is

needed before choosing the amount of padding used for RACF. A downside of the RACF method is that it can still require a large amount of memory for larger pad sizes. Therefore, we prefer the proximal gradient method because it is more efficient from a memory perspective. In the next section, we show recognition performance on the AT&T/ORL dataset using both methods to illustrate these points further.
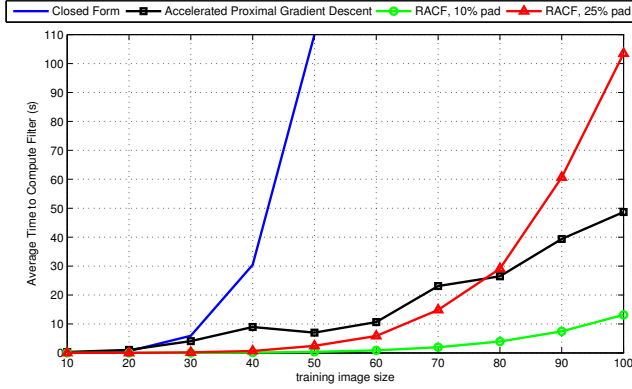

Fig. 8. Computational times for filter design comparing RACF and the proximal gradient method compared to a closed form solution. The "training image size" on the horizontal axis refers to N where the training image size is $N \times N$.

6 RECOGNITION PERFORMANCE

In this section, we report recognition performance of ZACFs. We emphasize that the ZA constraints introduced in this paper are a *fundamental* and *significant* improvement to the formulation of CFs. We have observed that ZACFs consistently perform better than their original formulation (aliased) counterparts. This is because the original CF designs did not account for circular correlation and are fundamentally flawed. This paper's focus is to illustrate the performance improvement that is realized by adding ZA constraints to CF designs, and *not* to compare different types of CFs to each other. Further, we are also *not* claiming that CFs are the best approach for all object recognition problems. To investigate the performance of ZACFs, we apply them to four types of pattern recognition tasks, namely face recognition, automatic target recognition (ATR), eye localization and object detection.

6.1 Face Recognition

We apply CFs to two different face recognition datasets: the AT&T (ORL) Database of Faces [36] and the face recognition grand challenge (FRGC) dataset [41].

6.1.1 AT&T Database of Faces

The AT&T Database of Faces (formerly the ORL Database of Faces) [36] contains face images of size 112×92. There are 40 subjects with 10 images each. We test using a leave-one-out cross validation approach. For each experiment, 9 training images were used to train one CF for each subject. These filters were then tested on the remaining image for all subjects. This

resulted in a total of 40 test images (one from each class) that were tested with 40 filters each, for a total of 1600 correlations per cross validation step. This was repeated 10 times. For each correlation output, we calculate peak to correlation energy (PCE), which is a measure of peak sharpness [42]. We perform classification on each of the test images and compute the rank-1 ID rate. We compute the ZACFs in several ways: 1) closed form solution, 2) accelerated proximal gradient descent, and 3) various versions of the RACF (using different amounts of zero-padding). The purpose of this comparison is to demonstrate the effectiveness of the computational solutions in Section 5. Note that computing the closed form solution is extremely difficult and requires many hours on high-end computers. Therefore we have not done an exhaustive comparison on every possible set of parameters available to each filter formulation. The results of our validation for the MACE, OTSDF, MOSSE, and MMCF filters are shown in Table 1. We have used a delta function for the desired correlation output for MOSSE and MMCF.

We observe that the ZACF outperforms the baseline (conventional) CFs. Note in Table 1 that the proximal gradient gives recognition performance close to that from the closed form solution. This is also true for RACF provided that the amount of zero-padding, q, is sufficiently large.

6.1.2 FRGC

The FRGC dataset [41] contains face images of resolution 128×128. We use 410 subjects from the test portion of the data, removing subjects with less than 8 images per subject. For the data we use, the number of images from each subject varies from 8 (minimum) to 88 (maximum) with a mean of 39 images per subject. We form three training and test sets by randomly selecting 25% of each class for training with the remaining 75% used for testing. We then build one CF per subject and apply every CF to every test image. Like the ORL dataset, we present our results in terms of equal error rate (EER) and Rank-1 ID rate in Table 2. Due to the training image size, we compute the ZACF with the accelerated proximal gradient approach for OTSDF, MACE, MOSSE, and MMCF. In the table, we refer to these results simply as "ZACF" for simplicity. As in the ORL experiments, we use a delta function for the desired correlation output, if required. Note that each ZACF achieves both a higher Rank-1 ID and a lower EER than the corresponding baseline filter.

TABLE 2. Performance Comparison of Baseline CFs and ZACFs on the FRGC Dataset

		Baseline CF	ZACF
OTSDF	EER	2.43%	1.79%
	Rank-1 ID	93.23%	94.95%
MACE	EER	15.23%	9.46%
	Rank-1 ID	52.73%	78.98%
MOSSE	EER	7.35%	5.02%
	Rank-1 ID	86.87%	93.78%
MMCF	EER	2.52%	1.80%
	Rank-1 ID	91.96%	94.52%

6.2 ATR Algorithm Development Image Database

We investigate vehicle recognition (i.e., simultaneous classification and localization) using a set of infrared images (frames

TABLE 1. Performance Comparison of Baseline CFs and ZACFs on the ORL Dataset

		Baseline CF	ZACF (closed form)	ZACF (proximal gradient)	RACF ($p = 15$)	RACF ($p = 20$)	RACF ($p = 25$)	RACF ($p = 30$)
OTSDF	EER	10.89%	7.54%	7.65%	11.75%	8.45%	7.21%	7.5%
	Rank-1 ID	83.75%	89.5%	89.5%	70.75%	81%	85.75%	87.75%
MOSSE	EER	11.75%	7.94%	7.94%	12.25%	9.25%	8.75%	8.5%
	Rank-1 ID	84.25%	88.25%	88%	72.75%	83%	85.75%	87.5%
MMCF	EER	11.73%	8%	8.21%	12%	8.930%	8.75%	8.5%
	Rank-1 ID	84.25%	88.5%	88.3%	73%	83%	86%	87.5%

from videos) from the *ATR Algorithm Development Image Database* [43]. This database contains infrared videos of 512×640 pixels/frame of eight vehicles (one per video), shown in Fig. 9, taken at multiple ranges during day and night at 30 fps. Note in Fig. 9 that some of the vehicles have very similar appearance, making the classification task challenging. In the dataset, these vehicles are driven at ~ 5 m/s, making a full circle of diameter of about 100 m, therefore exhibiting 360° of azimuth rotation. Each video is 1 minute long, allowing the vehicle to complete at least one full circle. An example frame is shown in Fig. 10. Note that the low quality frame and the general background makes the recognition task challenging. We design one template per vehicle (the template size was

Fig. 10. Vehicle "Pickup" and background

(a) Pickup (b) SUV (c) BTR70 (d) BRDM2

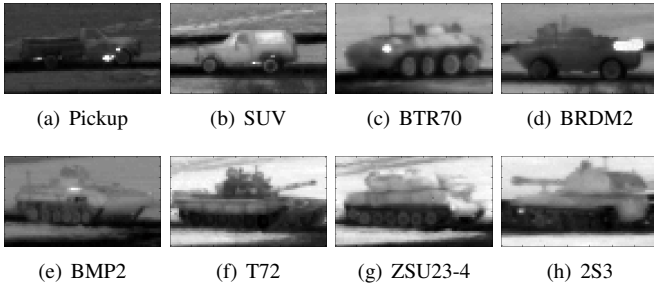(e) BMP2 (f) T72 (g) ZSU23-4 (h) 2S3

Fig. 9. Example images from different classes of vehicles.

small enough to easily calculate the closed form ZACFs) to classify the vehicle in the presence of 360° azimuth variation. We select 20 positive-class images per filter (manually cropped from the corresponding frames, $N \times M = 40 \times 70$) and 80 non-overlapping background images as negative class images for training. We select 200 full frames for testing, verifying that none of the testing frames were used in training. For testing, we correlate the 8 templates (one per vehicle) with each test image. For each correlation plane, we select the highest value, compute PCE, and assign the test image to the class that gives the highest PCE. If it is the correct class, we declare it as correctly classified. We declare correct localization when the location of the peak from the true class filter is within 20 and 35 pixels (i.e., half the size of the template) of the ground truth location in the vertical and horizontal directions, respectively. Finally, we declare a correct recognition when there is *both* correct classification *and* correct localization.

Table 3 shows the average classification, localization, and recognition percentages of both the conventional CF and ZACF. Dalal and Triggs [44] proposed cross-correlating the 2-D template with full training frames, adding the false positives as negative class training images, and retraining the template. We show our results after retraining in Table 4. We observe

that retraining helps all filters, but especially the MMCF filter, whose inequality constraints allow unlimited number of training images from two classes. The most important observation is that ZACFs always perform better than or about the same as traditional CFs in classification, localization, and recognition. Note that we treat each frame independently of other frames, i.e., although using a tracker would improve results, the purpose of these experiments is to compare the performance of conventional CFs and ZACFs without additional help from a tracker.

TABLE 3. CF Recognition Performance without Retraining, ATR Algorithm Development Image Database

	Classification		Localization		Recognition	
	Base	ZA	Base	ZA	Base	ZA
MACE	40.0%	51.9%	84.1%	88.9%	36.2%	47.5%
OTSDF	53.6%	62.1%	90.3%	90.3%	52.4%	57.4%
MOSSE	29.7%	32.3%	64.5%	78.1%	26.1%	30.6%
MMCF	51.1%	57.0%	87.7%	90.0%	49.6%	52.6%

TABLE 4. CF Recognition Performance with Retraining, ATR Algorithm Development Image Database

	Classification		Localization		Recognition	
	Base	ZA	Base	ZA	Base	ZA
MACE	46.2%	57.7%	85.1%	89.1%	42.7%	51.9%
OTSDF	59.9%	62.3%	90.7%	90.0%	58.1%	58.1%
MOSSE	31.9%	33.4%	76.5%	84.4%	31.1%	32.0%
MMCF	63.9%	74.3%	95.3%	96.1%	63.2%	73.5%

6.3 Eye Localization

Accurate localization of the eyes in face images is an important component of face, ocular, and iris recognition. In this

TABLE 5. Eye Localization Performance (%)

Eye	MOSSE	ZAMOSSE	MMCF	ZAMMCF
Right	94.66	97.24	95.01	97.14
Left	96.23	96.30	96.44	97.16

experiment we consider the task of accurately determining the location of the left and the right eye given a bounding box around a face obtained from a face detector. Since a good face detector makes eye localization overly simple, following the experimental setup outlined in [13], we make the problem more challenging by introducing errors in face localization. We first center the faces obtained using the *OpenCV* face detector to produce 128×128 images with the eyes centered at $(32.0, 40.0)$ and $(96.0, 40.0)$. We then apply a random similarity transform with translation of up to ± 4 pixels, scale factor of up to 1.0 ± 0.1, and rotations of up to $\frac{\pi}{16}$ radians. We used the FERET [45] database for this task, which has about 3400 images of 1204 people. We randomly partitioned the database with 512 images used for training, 675 for parameter selection by cross-validation, and the rest for testing. The CFs are compared by evaluating the normalized distance defined as,

$$D = \frac{\left\| P - \hat{P} \right\|}{\| P_l - P_r \|} \tag{30}$$

where P is the ground truth location, $\hat{P}$ is the predicted location, and P_l and P_r are the ground truth locations of the left and the right eye, respectively. The point $D = 0.1$ corresponds to detecting an object that is approximately the size of a human iris. We train MOSSE, MMCF, ZAMOSSE and ZAMMCF filters using 64×64 image patches centered at the left and right eye regions. We note that the MACE/OTSDF design is not suitable for this task since the number of training images is greater than the degrees of freedom in the filter. In such cases MACE/OTSDF designs result in overdetermined linear system and cannot be solved for. Therefore for this experiment we do not show results with MACE and OTSDF designs. We use the proximal gradient descent based algorithm to learn the ZA versions of MOSSE and MMCF. We evaluate the eye localization performance of the CF templates by searching over the entire face image i.e., the scenario where the approximate eye location is not known a-priori. We average the results over 5 different runs with random partitions for training and testing and random similarity transforms. We compare the eye localization performance as a function of D in Fig. 11 and Table 5 shows the performance of the MOSSE and MMCF filters along with their ZA versions at the operating point $D = 0.1$. The filters, while producing a strong response for the correct eye, are sometimes distracted by the wrong eye or other parts of the face. Note that the ZA filters provide a higher localization accuracy than the filters with aliasing. The absence of aliasing results in lower noise levels in the correlation outputs thereby resulting in better localization performance.

6.4 Object Detection

We evaluated various CF and ZACF designs for pedestrian detection on the INRIA Pedestrians [44] dataset and the
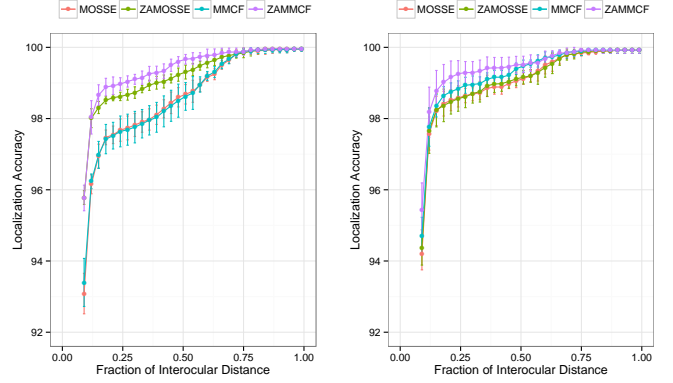

Fig. 11. Eye localization performance of the left (left) and right (right) eye, respectively, as a function of the inter-ocular distance for the different CF designs. We show the mean performance across five random runs along with the error bars. The ZA filters outperform those designed without explicitly accounting for aliasing.

ETHZ Shapes dataset [46]. Instead of mining for hard negatives, as is commonly done, we simply consider a large set of negative windows following the protocol in [6]. For this set of experiments we use vector-valued HOG feature representation [44], i.e., we have $K = 36$ different feature channels. The ETHZ Shapes dataset consists of 5 categories (Mug, Bottle, Swan, Giraffe and Apple Logo) each consisting of between 22 and 45 positive samples. We compare the following different CFs: Vector Correlation Filter (VCF) [3], Maximum-Margin Vector Correlation Filter (MMVCF) [7], Circulant Decomposition (CD) [6] based CF, and the ZA filters ZAVCF (Eq. 16), ZAMMVCF (Eq. 27). In Table 6 we present the average precision for the INRIA dataset and the mean average precision over all the ETHZ Shapes and Fig. 12 and Fig.13 show the precision-recall curves for the INRIA pedestrian dataset and each object in the ETHZ Shapes dataset respectively. We note that among single linear template based methods, ZAMMVCF results in the best performance on both these datasets.

TABLE 6. Object Detection: Average Precision (%)

Dataset	VCF [3]	ZAVCF	MMVCF [7]	ZAMMVCF	CD [6]
INRIA	79.79	80.81	83.19	84.07	79.56
ETHZ	76.74	76.70	78.33	80.10	78.16

7 CONCLUSIONS

In this paper, we proposed and investigated a fundamental and significant advancement to the design of CFs. Existing CF designs that are formulated in the frequency domain do not explicitly account for the fact that multiplication of two DFTs in the frequency domain corresponds to a circular correlation in the spatial domain. As a result, existing CF designs do not actually optimize their intended cost functions. In this paper, we present a solution to this problem (ZACFs) that completely removes circular correlation effects from CF designs. While we have explicitly shown new filter derivations for several popular filter designs, this approach can be used with many more
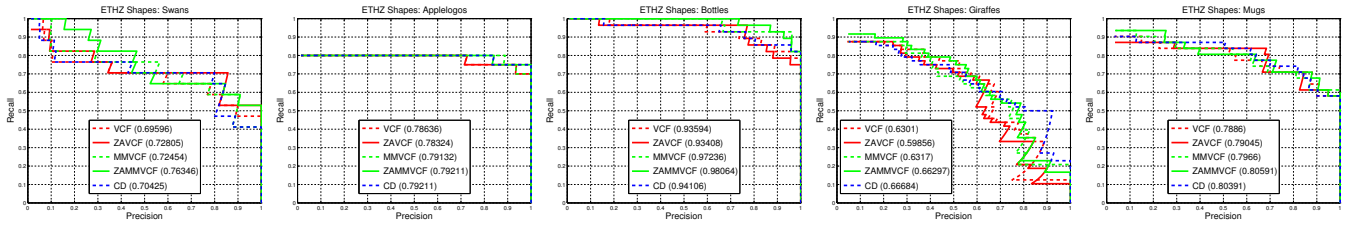
Fig. 13. Comparison of various correlation filter designs for object shape detection on the ETHZ Shapes dataset.
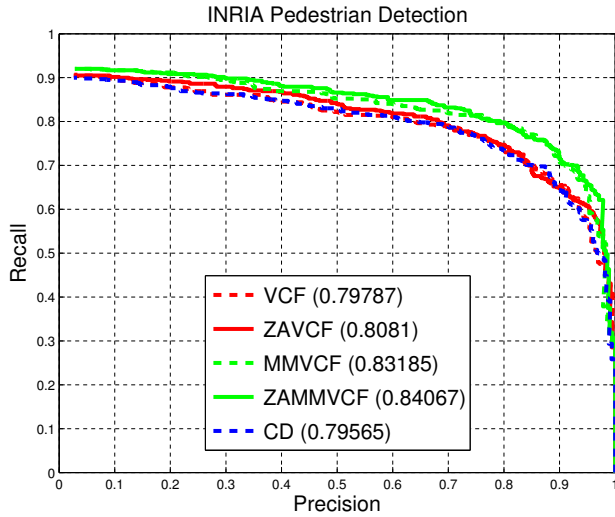

Fig. 12. Comparison of various correlation filter designs for pedestrian detection on the INRIA Pedestrian dataset.

CFs and with other approaches. To address the computational challenges caused by the ZA constraints we introduced the RACF designs as an approximate solution as well as proximal gradient descent based algorithms for exactly solving for the various ZACFs. We have shown that our methods eliminate aliasing and lead to significantly better results across the board for many different CFs and for different datasets.

ACKNOWLEDGMENT

REFERENCES

[1] B. V. K. Vijaya Kumar, M. Savvides, and C. Xie, "Correlation pattern recognition for face recognition," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1963–1976, 2006.

[2] J. Thornton, M. Savvides, and B. V. K. Vijaya Kumar, "A Bayesian approach to deformed pattern matching of iris images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 596–606, 2007.

[3] V. N. Boddeti, T. Kanade, and B. V. K. Vijaya Kumar, "Correlation filters for object alignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2291–2298, 2013.

[4] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[5] A. Rodriguez, V. N. Boddeti, B. V. K. Vijaya Kumar, and A. Mahalanobis, "Maximum margin correlation filter: A new approach for localization and classification," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 631–643, 2013.

[6] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista, "Beyond hard negative mining: Efficient detector learning via block-circulant decomposition," in *IEEE International Conference on Computer Vision*, pp. 2760–2767, 2013.

[7] V. N. Boddeti and B. V. K. Vijaya Kumar, "Maximum margin vector correlation filter," in *Arxiv 1404.6031*, 2014.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*, pp. 702–715, Springer, 2012.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010.

[10] J. Revaud, M. Douze, C. Schmid, and H. Jégou, "Event retrieval in large video collections with circulant temporal encoding," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2459–2466, 2013.

[11] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. Pearson Prentice Hall, 2007.

[12] A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, no. 17, pp. 3633–3640, 1987.

[13] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2105–2112, 2009.

[14] B. V. K. Vijaya Kumar, A. Mahalanobis, and D. W. Carlson, "Optimal trade-off synthetic discriminant function filters for arbitrary devices," *Optics Letters*, vol. 19, no. 19, pp. 1556–1558, 1994.

[15] D. Casasent, G. Ravichandran, and S. Bollapragada, "Gaussian–minimum average correlation energy filters," *Applied Optics*, vol. 30, no. 35, pp. 5176–5181, 1991.

[16] P. Refregier, "Filter design for optical pattern recognition: multicriteria optimization approach," *Optics Letters*, vol. 15, no. 15, pp. 854–856, 1990.

[17] J. A. Fernandez and B. V. K. Vijaya Kumar, "Zero-aliasing correlation filters," *International Symposium on Image and Signal Processing and Analysis*, pp. 101–106, 2013.

[18] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," pp. 391–398, 2013.

[19] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through FFTs," *International Conference on Learning Representations*, 2014.

[20] V. N. Boddeti, J. M. Smereka, and B. V. Kumar, "A comparative evaluation of iris and ocular recognition methods on challenging ocular images," in *International Joint Conference on Biometrics*, IEEE, 2011.

[21] V. N. Boddeti and B. V. K. Vijaya Kumar, "A framework for binding and retrieving class-specific information to and from image patterns using correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2064–2077, 2013.

[22] K. Miyazawa, K. Ito, T. Aoki, K. Kobayashi, and H. Nakajima, "An effective approach for iris recognition using phase-based image matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1741–1756, 2008.

[23] H. Nakajima, K. Kobayashi, and T. Higuchi, "A fingerprint matching algorithm using phase-only correlation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 87, no. 3, pp. 682–691, 2004.

[24] K. Ito and T. Aoki, "Phase-based image matching and its application to biometric recognition," *Proceedings of APSIPA Annual Summit and Conference*, 2013.

[25] R. Kerekes and B. V. K. Vijaya Kumar, "Enhanced video-based target detection using multi-frame correlation filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 289–307, 2009.

[26] D. S. Bolme, Y. M. Lui, B. A. Draper, and J. R. Beveridge, "Simple real-time human detection using a single correlation filter," in *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.

[27] Y. Li, Z. Wang, and H. Zeng, "Correlation filter: An accurate approach to detect and locate low contrast character strings in complex table environment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1639–1644, 2004.

[28] Y. Movshovitz-Attias, V. N. Boddeti, Z. Wei, and Y. Sheikh, "3d pose-by-detection of vehicles via discriminatively reduced ensembles of correlation filters," in *British Machine Vision Conference*, 2014.

[29] S. Ali and S. Lucey, "Are correlation filters useful for human action recognition?," in *International Conference on Pattern Recognition*, pp. 2608–2611, 2010.

[30] A. Mahalanobis, R. Stanfill, and K. Chen, "A bayesian approach to activity detection in video using multi-frame correlation filters," *Proceedings of SPIE*, vol. 8049, 2011.

[31] A. Haq, I. Gondal, and M. Murshed, "Action recognition using spatio-temporal distance classifier correlation filter," in *International Conference on Digital Image Computing Techniques and Applications*, pp. 474–479, 2011.

[32] T. S. H. Kiani Galoogahi and S. Lucey, "Multi-channel correlation filters," *IEEE International Conference on Computer Vision*, pp. 4321–4328, 2013.

[33] S. I. Sudharsanan, A. Mahalanobis, and M. K. Sundareshan, "Unified framework for the synthesis of synthetic discriminant functions with reduced noise variance and sharp correlation structure," *Optical Engineering*, vol. 29, no. 9, pp. 1021–1028, 1990.

[34] A. Rodriguez and B. V. K. Vijaya Kumar, "Dealing with circular correlation effects," in *SPIE Conference on Automatic Target Recognition*, vol. 8744, 2013.

[35] G. Moody and R. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[36] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *IEEE Workshop on Applications of Computer Vision*, pp. 138–142, 1994.

[37] B. V. K. Vijaya Kumar, A. Mahalanobis, S. Song, S. R. F. Sims, and J. F. Epperson, "Minimum squared error synthetic discriminant functions," *Optical Engineering*, vol. 31, no. 5, pp. 915–922, 1992.

[38] A. Rodriguez, *Maximum Margin Correlation Filters*. PhD Thesis, Carnegie Mellon University, 2012.

[39] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[41] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 947–954, 2005.

[42] B. V. K. Vijaya Kumar, A. Mahalanobis, and R. D. Juday, *Correlation Pattern Recognition*. Cambridge University Press, 2005.

[43] Military Sensing Information Analysis Center, "ATR algorithm development image database." www.sensiac.org, 2011.

[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.

[45] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[46] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.
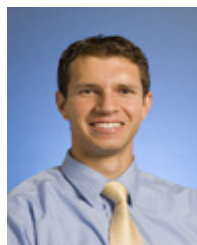
Joseph A. Fernandez received his BS in Electrical Engineering from New Mexico Tech in 2008 and his MS and PhD in Electrical and Computer Engineering from Carnegie Mellon University (CMU) in 2010 and 2014, respectively. His research interests include correlation filters and improving the efficiency of signal processing algorithms. He is a 2010 National Defense Science and Engineering Graduate (NDSEG) Fellow and received the Best Paper Gold Award for a student/young researcher at the International Symposium on Image and Signal Processing and Analysis in 2013.

Vishnu Naresh Boddeti received a BTech degree in Electrical Engineering from the Indian Institute of Technology, Madras in 2007, and his MSc and Ph.D. degree in Electrical and Computer Engineering program at Carnegie Mellon University. He is currently at postdoc at the Robotics Institute at Carnegie Mellon University. His research interests are in Computer Vision, Pattern Recognition and Machine Learning. He was awarded the Carnegie Institute of Technology Dean's Tuition Fellowship in 2007 and received the best paper award at the BTAS conference in 2013.

Andres Rodriguez is a research scientist at the Air Force Research Laboratory Sensors Directorate. His research interests are in pattern recognition, machine learning, and computer vision. He received a BS and MS degree in Electrical Engineering from Brigham Young University in 2006 and 2008, respectively, and his PhD in Electrical and Computer Engineering from Carnegie Mellon University in 2012. He was awarded the Carnegie Institute of Technology Dean's Tuition Fellowship in 2008 and the Frank J. Marshall Graduate Fellowship in 2009. He has published over a dozen technical papers and received the best student paper award at the SPIE ATR conference in 2010.

B. V. K. Vijaya Kumar is the U.A. & Helen Whitaker Professor of Electrical and Computer Engineering (ECE) and the Associate Dean for Graduate and Faculty Affairs of College of Engineering at Carnegie Mellon University (CMU). Professor Kumar served as the Interim Dean of College and Engineering at CMU from 2012 to 2013. Professor Kumar's research interests include Pattern Recognition, Biometrics and Coding and Signal Processing for Data Storage Systems. Professor Kumar is a Fellow of IEEE, SPIE, OSA and IAPR. His publications include the book entitled Correlation Pattern Recognition, twenty book chapters and more than 550 technical papers. He served as a Pattern Recognition Topical Editor for Applied Optics and as an Associate Editor for IEEE Transactions on Information Forensics and Security. Prof. Kumar is a co-recipient of the 2009 Carnegie Institute of Technology's Outstanding Faculty Research Award.