Correlation Filters with Limited Boundaries

Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey
{*hkiani, tsim*}*@comp.nus.edu.sg, simon.lucey@csiro.au*

Abstract—Correlation filters take advantage of specific properties in the Fourier domain allowing them to be estimated efficiently: $\mathcal{O}(ND \log D)$ in the frequency domain, versus $\mathcal{O}(D^3 + ND^2)$ spatially where D is signal length, and N is the number of signals. Recent extensions to correlation filters, such as MOSSE, have reignited interest of their use in the vision community due to their robustness and attractive computational properties. In this paper we demonstrate, however, that this computational efficiency comes at a cost. Specifically, we demonstrate that only $\frac{1}{D}$ proportion of shifted examples are unaffected by boundary effects which has a dramatic effect on detection/tracking performance. In this paper, we propose a novel approach to correlation filter estimation that: (i) takes advantage of inherent computational redundancies in the frequency domain, and (ii) dramatically reduces boundary effects. Impressive object tracking and detection results are presented in terms of both accuracy and computational efficiency.

Index Terms—Correlation filters, object tracking, pattern detection

1 Introduction

Correlation between two signals is a standard approach to feature detection/matching. Correlation touches nearly every facet of computer vision from pattern detection to object tracking. Correlation is rarely performed naively in the spatial domain. Instead, the fast Fourier transform (FFT) affords the efficient application of correlating a desired template/filter with a signal. Contrastingly, however, most techniques for estimating a template for such a purpose (i.e. detection/tracking through convolution) are performed in the spatial domain [1], [2], [16], [18].

This has not always been the case. Correlation filters, developed initially in the seminal work of Hester and Casasent [12], are a method for learning a template/filter in the frequency domain that rose to some prominence in the 80s and 90s. Although many variants have been proposed [12], [13], [15], [14], the approach's central tenet is to learn a filter, that when correlated with a set of training signals, gives a desired response (typically a peak at the origin of the object, with all other regions of the correlation response map being suppressed). Like correlation itself, one of the central advantages of the approach is that it attempts to learn the filter in the frequency domain due to the efficiency of correlation/convolution in that domain.

Interest in correlation filters has been reignited in the vision world through the recent work of Bolme et al. [4] on Minimum Output Sum of Squared Error (MOSSE) correlation filters for object detection and tracking. Bolme et al.'s work was able to circumvent some of the classical problems with correlation filters and performed well in tracking under changes in rotation, scale, lighting and partial occlusion. A central strength of the correlation filter is that it is extremely efficient in terms of both memory and computation.

1.1 The Problem

An unconventional interpretation of a correlation filter, is that of a discriminative template that has been estimated from

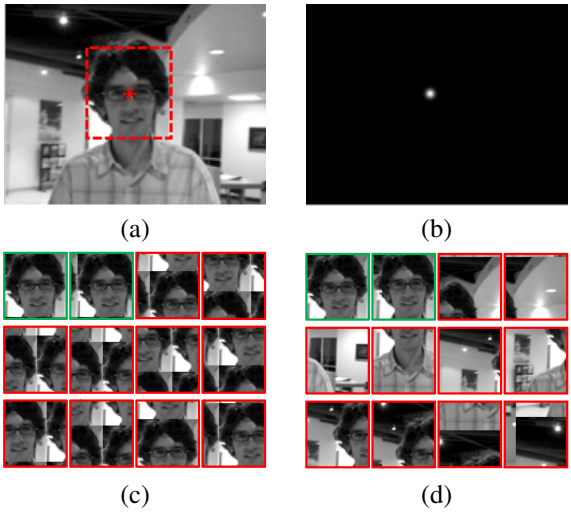


Fig. 1. (a) Defines the example of fixed spatial support within the image from which the peak correlation output should occur. (b) The desired output response, based on (a), of the correlation filter when applied to the entire image. (c) A subset of patch examples used in a canonical correlation filter where green denotes a non-zero correlation output, and red denotes a zero correlation output in direct accordance with (b). (d) A subset of patch examples used in our proposed correlation filter. Note that our proposed approach uses patches stemming from different parts of the image, whereas the canonical correlation filter simply employs circular shifted versions of the same single patch. The central dilemma in this paper is how to perform (d) efficiently in the Fourier domain. The two last patches of (d) show that $\frac{D-1}{T}$ patches near the image border are affected by circular shift in our method which can be greatly diminished by choosing $D << T$, where D and T indicate the length of the vectorized face patch in (a) and the image in (a), respectively.

an unbalanced set of "real-world" and "synthetic" examples. These synthetic examples are created through the application of a circular shift on the real-world examples, and are supposed to be representative of those examples at different translational shifts. We use the term synthetic, as all these shifted examples are plagued by circular boundary effects and are not truly representative of the shifted example (see Figure 1(c)). As a result the training set used for learning the template is extremely unbalanced with one real-world example for every $D-1$ synthetic examples (where D is the dimensionality of the examples).

These boundary effects can dramatically affect the resulting performance of the estimated template. Fortunately, these effects can be largely removed (see Section 2) if the correlation filter objective is slightly augmented, but has to be now solved in the spatial rather than frequency domains. Unfortunately, this shift to the spatial domain destroys the computational efficiency that make correlation filters so attractive. It is this dilemma that is at the heart of our paper.

1.2 Contribution

In this paper we make the following contributions:

- We propose a new correlation filter objective that can drastically reduce the number of examples in a correlation filter that are affected by boundary effects. We further demonstrate, however, that solving this objective in closed form drastically decreases computational efficiency: $\emptyset(D^3 + ND^2)$ versus $\emptyset(ND\log D)$ for the canonical objective where D is the length of the vectorized image and N is the number of examples.
- We demonstrate how this new objective can be efficiently optimized in an iterative manner through an Augmented Lagrangian Method (ALM) so as to take advantage of inherent redundancies in the frequency domain. The efficiency of this new approach is $\emptyset([N+K]T\log T)$ where K is the number of iterations and T is the size of the search window.
- We present impressive results for both object detection and tracking outperforming MOSSE and other leading non-correlation filter methods for object tracking.

1.3 Related Work

Bolme et al. [4] recently proposed an extension to traditional correlation filters referred to as Minimum Output Sum of Squared Error (MOSSE) filters. This approach has proven invaluable for many object tracking tasks, outperforming current state of the art methods such as [2], [18]. What made the approach of immediate interest in the vision community was the dramatically faster frame rates than current state of the art (600 fps versus 30 fps). A strongly related method to MOSSE was also proposed by Bolme et al. [5] for object detection/localization referred to as Average of Synthetic Exact Filters (ASEF) which also reported superior performance to state of the art. A full discussion on other variants of correlation filters such as Optimal Tradeoff Filters (OTF) [17], Unconstrained MACE (UMACE) [19] filters, etc. is outside the scope of this paper. Readers are encouraged to inspect [14] for a full treatment on the topic.

1.4 Notation

Vectors are always presented in lower-case bold (e.g., $\mathbf{a}$), Matrices are in upper-case bold (e.g., $\mathbf{A}$) and scalars in italicized (e.g. a or A). $\mathbf{a}(i)$ refers to the ith element of the vector $\mathbf{a}$. All M-mode array signals shall be expressed in vectorized form $\mathbf{a}$. M-mode arrays are also known as M-mode matrices, multidimensional matrices, or tensors. We shall be assuming $M=2$ mode matrix signals (e.g. $2D$ image arrays) in nearly all our discussions throughout this paper. This does not preclude, however, the application of our approach to other $M \neq 2$ signals.

A M-mode convolution operation is represented as the $*$ operator. One can express a M-dimensional discrete circular shift $\Delta\boldsymbol{\tau}$ to a vectorized M-mode matrix $\mathbf{a}$ through the notation $\mathbf{a}[\Delta\boldsymbol{\tau}]$. The matrix $\mathbf{I}$ denotes a $D\times D$ identity matrix and $\mathbf{1}$ denotes a D dimensional vector of ones. A $\hat{}$ applied to any vector denotes the M-mode Discrete Fourier Transform (DFT) of a vectorized M-mode matrix signal $\mathbf{a}$ such that $\hat{\mathbf{a}} \leftarrow \mathcal{F}(\mathbf{a}) = \sqrt{D}\mathbf{F}\mathbf{a}$. Where $\mathcal{F}()$ is the Fourier transforms operator and $\mathbf{F}$ is the orthonormal $D\times D$ matrix of complex basis vectors for mapping to the Fourier domain for any D dimensional vectorized image/signal. We have chosen to employ a Fourier representation in this paper due to its particularly useful ability to represent circular convolutions as a Hadamard product in the Fourier domain. Additionally, we take advantage of the fact that $\text{diag}(\hat{\mathbf{h}})\hat{\mathbf{a}} = \hat{\mathbf{h}} \circ \hat{\mathbf{a}}$, where $\circ$ represents the Hadamard product, and diag() is an operator that transforms a D dimensional vector into a $D\times D$ dimensional diagonal matrix. The role of filter $\hat{\mathbf{h}}$ or signal $\hat{\mathbf{a}}$ can be interchanged with this property. Any transpose operator $^\top$ on a complex vector or matrix in this paper additionally takes the complex conjugate in a similar fashion to the Hermitian adjoint [14]. The operator $\text{conj}(\hat{\mathbf{a}})$ applies the complex conjugate to the complex vector $\hat{\mathbf{a}}$.

2 Correlation Filters

Due to the efficiency of correlation in the frequency domain, correlation filters have canonically been posed in the frequency domain. There is nothing, however, stopping one (other than computational expense) from expressing a correlation filter in the spatial domain. In fact, we argue that viewing a correlation filter in the spatial domain can give: (i) important links to existing spatial methods for learning templates/detectors, and (ii) crucial insights into fundamental problems in current correlation filter methods.

Bolme et. al's [4] MOSSE correlation filter can be expressed in the spatial domain as solving the following ridge regression problem,

$$E(\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D}||\mathbf{y}_i(j) - \mathbf{h}^\top\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \quad (1)$$

where $\mathbf{y}_i \in \mathbb{R}^D$ is the desired response for the i-th observation $\mathbf{x}_i \in \mathbb{R}^D$ and λ is a regularization term. $\mathbb{C} = [\Delta\boldsymbol{\tau}_1, \ldots, \Delta\boldsymbol{\tau}_D]$ represents the set of all circular shifts for a signal of length D. Bolme et al. advocated the use of a 2D Gaussian of small variance (2-3 pixels) for $\mathbf{y}_i$ centered at the

location of the object (typically the centre of the image patch). The solution to this objective becomes,

$$\mathbf{h} = \mathbf{H}^{-1} \sum_{i=1}^{N} \sum_{j=1}^{D} \mathbf{y}_i(j) \mathbf{x}_i[\Delta\boldsymbol{\tau}_j] \tag{2}$$

where,

$$\mathbf{H} = \lambda \mathbf{I} + \sum_{i=1}^{N} \sum_{j=1}^{D} \mathbf{x}_i[\Delta\boldsymbol{\tau}_j] \mathbf{x}_i[\Delta\boldsymbol{\tau}_j]^\top \quad . \tag{3}$$

Solving a correlation filter in the spatial domain quickly becomes intractable as a function of the signal length D, as the cost of solving Equation 2 becomes $\mathcal{O}(D^3 + ND^2)$.

2.1 Properties

Putting aside, for now, the issue of computational cost, the correlation filter objective described in Equation 1 produces a filter that is particularly sensitive to misalignment in translation. A highly undesirable property when attempting to detect or track an object in terms of translation. This sensitivity is obtained due to the circular shift operator $\mathbf{x}[\Delta\boldsymbol{\tau}]$, where $\Delta\boldsymbol{\tau} = [\Delta x, \Delta y]^\top$ denotes the 2D circular shift in x and y.

It has been well noted in correlation filter literature [14] that this circular-shift alone tends to produce filters that do not generalize well to other types of appearance variation (e.g. illumination, viewpoint, scale, rotation, etc.). This generalization issue can be somewhat mitigated through the judicious choice of non-zero regularization parameter λ, and/or through the use of an ensemble $N > 1$ of training observations that are representative of the type of appearance variation one is likely to encounter.

2.2 Boundary Effects

A deeper problem with the objective in Equation 1, however, is that the shifted image patches $\mathbf{x}[\Delta\boldsymbol{\tau}]$ at all values of $\Delta\boldsymbol{\tau} \in \mathbb{C}$, except where $\Delta\boldsymbol{\tau} = \mathbf{0}$, are not representative of image patches one would encounter in a normal correlation operation (Figure 1(c)). In signal-processing, one often refers to this as the *boundary effect*. One simple way to circumvent this problem spatially is to allow the training signal $\mathbf{x} \in \mathbb{R}^T$ to be a larger size than the filter $\mathbf{h} \in \mathbb{R}^D$ such that $T > D$. Through the use of a $D \times T$ masking matrix $\mathbf{P}$ one can reformulate Equation 1 as,

$$E(\mathbf{h}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{T} ||\mathbf{y}_i(j) - \mathbf{h}^\top \mathbf{P} \mathbf{x}_i[\Delta\boldsymbol{\tau}_j]||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2 \quad . \tag{4}$$

The masking matrix $\mathbf{P}$ of ones and zeros encapsulates what part of the signal should be active/inactive. The central benefit of this augmentation in Equation 4 is the dramatic increase in the proportion of examples unaffected by boundary effects ($\frac{T-D+1}{T}$ instead of $\frac{1}{D}$). From this insight it becomes clear that if one chooses $T >> D$ then boundary effects become greatly diminished (Figure 1(d)). The computational cost $\mathcal{O}(D^3 + NTD)$ of solving this objective is only slightly larger than the cost of Equation 1, as the role of $\mathbf{P}$ in practice can be accomplished efficiently through a lookup table.

It is clear in Equation 4, that boundary effects could be removed completely by summing over only a $T - D + 1$ subset of all the T possible circular shifts. However, as we will see in the following section such a change along with the introduction of $\mathbf{P}$ is not possible if we want to solve this objective efficiently in the frequency domain.

2.3 Efficiency in the Frequency Domain

It is well understood in signal processing that circular convolution in the spatial domain can be expressed as a Hadamard product in the frequency domain. This allows one to express the objective in Equation 1 more succinctly and equivalently as,

$$\begin{aligned} E(\hat{\mathbf{h}}) &= \frac{1}{2} \sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{h}})||_2^2 + \frac{\lambda}{2} ||\hat{\mathbf{h}}||_2^2 \\ &= \frac{1}{2} \sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{h}}||_2^2 + \frac{\lambda}{2} ||\hat{\mathbf{h}}||_2^2 \quad . \end{aligned} \tag{5}$$

where $\hat{\mathbf{h}}, \hat{\mathbf{x}}, \hat{\mathbf{y}}$ are the Fourier transforms of $\mathbf{h}, \mathbf{x}, \mathbf{y}$. The complex conjugate of $\hat{\mathbf{h}}$ is employed to ensure the operation is correlation not convolution. The equivalence between Equations 1 and 5 also borrows heavily upon another well known property from signal processing namely, Parseval's theorem which states that

$$\mathbf{x}_i^\top \mathbf{x}_j = D^{-1} \hat{\mathbf{x}}_i^\top \hat{\mathbf{x}}_j \quad \forall i, j, \quad \text{where } \mathbf{x} \in \mathbb{R}^D \quad . \tag{6}$$

The solution to Equation 5 becomes

$$\begin{aligned} \hat{\mathbf{h}} &= [\text{diag}(\hat{\mathbf{s}}_{xx}) + \lambda \mathbf{I}]^{-1} \sum_{i=1}^{N} \text{diag}(\hat{\mathbf{x}}_i) \hat{\mathbf{y}}_i \\ &= \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1}) \end{aligned} \tag{7}$$

where $\circ^{-1}$ denotes element-wise division, and

$$\hat{\mathbf{s}}_{xx} = \sum_{i=1}^{N} \hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{x}}_i) \quad \& \quad \hat{\mathbf{s}}_{xy} = \sum_{i=1}^{N} \hat{\mathbf{y}}_i \circ \text{conj}(\hat{\mathbf{x}}_i) \tag{8}$$

are the average auto-spectral and cross-spectral energies respectively of the training observations. The solution for $\hat{\mathbf{h}}$ in Equations 1 and 5 are identical (other than that one is posed in the spatial domain, and the other is in the frequency domain). The power of this method lies in its computational efficiency. In the frequency domain a solution to $\hat{\mathbf{h}}$ can be found with a cost of $\mathcal{O}(ND \log D)$. The primary cost is associated with the DFT on the ensemble of training signals $\{\mathbf{x}_i\}_{i=1}^N$ and desired responses $\{\mathbf{y}_i\}_{i=1}^N$.

3 Our Approach

A problem arises, however, when one attempts to apply the same Fourier insight to the augmented spatial objective in Equation 4,

$$E(\mathbf{h}) = \frac{1}{2} \sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \sqrt{D} \mathbf{F} \mathbf{P}^\top \mathbf{h}||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2 \quad . \tag{9}$$

Unfortunately, since we are enforcing a spatial constraint the efficiency of this objective balloons to $\mathcal{O}(D^3 + ND^2)$ as $\mathbf{h}$ *must* be solved in the spatial domain.

3.1 Augmented Lagrangian

Our proposed approach for solving Equation 9 involves the introduction of an auxiliary variable $\hat{\mathbf{g}}$,

$$\begin{aligned} E(\mathbf{h}, \hat{\mathbf{g}}) &= \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{g}}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \\ &\text{s.t.} \quad \hat{\mathbf{g}} = \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h} \ . \end{aligned} \tag{10}$$

We propose to handle the introduced equality constraints through an Augmented Lagrangian Method (ALM) [6]. The augmented Lagrangian of our proposed objective can be formed as,

$$\begin{aligned} Ł(\hat{\mathbf{g}}, \mathbf{h}, \hat{\boldsymbol{\zeta}}) &= \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{g}}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \\ &+ \hat{\boldsymbol{\zeta}}^\top(\hat{\mathbf{g}} - \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}) \\ &+ \frac{\mu}{2}||\hat{\mathbf{g}} - \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}||_2^2 \end{aligned} \tag{11}$$

where μ is the penalty factor that controls the rate of convergence of the ALM, and $\hat{\boldsymbol{\zeta}}$ is the Fourier transform of the Lagrangian vector needed to enforce the newly introduced equality constraint in Equation 10. ALMs are not new to learning and computer vision, and have recently been used to great effect in a number of applications [6], [7]. Specifically, the Alternating Direction Method of Multipliers (ADMMs) has provided a simple but powerful algorithm that is well suited to distributed convex optimization for large learning and vision problems. A full description of ADMMs is outside the scope of this paper (readers are encouraged to inspect [6] for a full treatment and review), but they can be loosely interpreted as applying a Gauss-Seidel optimization strategy to the augmented Lagrangian objective. Such a strategy is advantageous as it often leads to extremely efficient subproblem decompositions. A full description of our proposed algorithm can be seen in Algorithm 1. We detail each of the subproblems as follows:

3.2 Subproblem g

$$\begin{aligned} \hat{\mathbf{g}}^* &= \arg\min Ł(\hat{\mathbf{g}}; \hat{\mathbf{h}}, \hat{\boldsymbol{\zeta}}) \\ &= (\hat{\mathbf{s}}_{xy} + \mu\hat{\mathbf{h}} - \hat{\boldsymbol{\zeta}}) \circ^{-1} (\hat{\mathbf{s}}_{xx} + \mu\mathbf{1}) \end{aligned} \tag{12}$$

where $\hat{\mathbf{h}} = \sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{h}$. In practice $\hat{\mathbf{h}}$ can be estimated extremely efficiently by applying a FFT to $\mathbf{h}$ padded with zeros implied by the $\mathbf{P}^\top$ masking matrix.

3.3 Subproblem h

$$\begin{aligned} \mathbf{h}^* &= \arg\min Ł(\mathbf{h}; \mathbf{g}, l) \\ &= (\mu + \frac{\lambda}{\sqrt{D}})^{-1}(\mu\mathbf{g} + l) \end{aligned} \tag{13}$$

where $\mathbf{g} = \frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\mathbf{g}}$ and $l = \frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\boldsymbol{\zeta}}$. In practice both $\mathbf{g}$ and l can be estimated extremely efficiently by applying an inverse FFT and then applying the lookup table implied by the masking matrix $\mathbf{P}$.

3.4 Lagrange Multiplier Update

$$\hat{\boldsymbol{\zeta}}^{(i+1)} \leftarrow \hat{\boldsymbol{\zeta}}^{(i)} + \mu(\hat{\mathbf{g}}^{(i+1)} - \hat{\mathbf{h}}^{(i+1)}) \tag{14}$$

where $\hat{\mathbf{g}}^{(i+1)}$ and $\hat{\mathbf{h}}^{(i+1)}$ are the current solutions to the above subproblems at iteration $i+1$ within the iterative ADMM.

3.5 Choice of μ

A simple and common [6] scheme for selecting μ is the following

$$\mu^{(i+1)} = \min(\mu_{\max}, \beta\mu^{(i)}) \ . \tag{15}$$

We found experimentally $\mu^{(0)} = 10^{-2}$, $\beta = 1.1$ and $\mu_{\max} = 20$ to perform well.

3.6 Computational Cost

Inspecting Algorithm 1 the dominant cost per iteration of the ADMM optimization process is $\emptyset(T\log T)$ for FFT. There is a pre-computation cost (before the iterative component, steps 4 and 5) in the algorithm for estimating the auto- and cross-spectral energy vectors $\hat{\mathbf{s}}_{xx}$ and $\hat{\mathbf{s}}_{xy}$ respectively. This cost is $\emptyset(NT\log T)$ where N refers to the number of training signals. Given that K represents the number of ADMM iterations the overall cost of the algorithm is therefore $\emptyset([N+K]T\log T)$.

Algorithm 1 Our approach using ADMMs

1: Intialize $\mathbf{h}^{(0)}$, $l^{(0)}$.
2: Pad with zeros and apply FFT: $\sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{h}^{(0)} \rightarrow \hat{\mathbf{h}}^{(0)}$.
3: Apply FFT: $\sqrt{D}\mathbf{F}l^{(0)} \rightarrow \hat{\boldsymbol{\zeta}}^{(0)}$.
4: Estimate auto-spectral energy $\hat{\mathbf{s}}_{xx}$ using Eqn. (8).
5: Estimate cross-spectral energy $\hat{\mathbf{s}}_{xy}$ using Eqn. (8).
6: $i = 0$
7: **repeat**
8: Solve for $\hat{\mathbf{g}}^{(i+1)}$ using Eqn. (12), $\hat{\mathbf{h}}^{(i)}$ & $\hat{\boldsymbol{\zeta}}^{(i)}$.
9: Inverse FFT then crop: $\frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\mathbf{g}}^{(i+1)} \rightarrow \mathbf{g}^{(i+1)}$.
10: Inverse FFT then crop: $\frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\boldsymbol{\zeta}}^{(i+1)} \rightarrow l^{(i+1)}$.
11: Solve for $\mathbf{h}^{(i+1)}$ using Eqn. (13), $\mathbf{g}^{(i+1)}$ & $l^{(i)}$.
12: Pad and apply FFT: $\sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{h}^{(i+1)} \rightarrow \hat{\mathbf{h}}^{(i+1)}$.
13: Update Lagrange multiplier vector Eqn. (14).
14: Update penalty factor Eqn. (15).
15: $i = i + 1$
16: **until** $\hat{\mathbf{g}}, \mathbf{h}, \hat{\boldsymbol{\zeta}}$ has converged

4 Experiments

4.1 Localization Performance

In the first experiment, we evaluated our method on the problem of eye localization, comparing with prior correlation filters, e.g. OTF [17], MACE [15], UMACE [19], ASEF [5], and MOSSE [4]. The CMU Multi-PIE face database [1] was used for this experiment, containing 900 frontal faces with neutral expression and normal illumination. We randomly

1. http://www.multipie.org/

selected 400 of these images for training and the reminder for testing. All images were cropped to have a same size of 128×128 such that the left and right eye are respectively centered at (40,32) and (40,96) coordinates. The cropped images were power normalized to have a zero-mean and standard deviation of 1.0. Then, a 2D cosine window was employed to reduce the frequency effects caused by opposite borders of the images in the Fourier domain.

We trained a 64×64 filter of the right eye using 64×64 cropped patches (centered upon the right eye) for the other methods, and full face images for our method ($T = 128 \times 128$ and $D = 64 \times 64$). Similar to ASEF and MOSSE, we defined the desired response as a 2D Gaussian function with an spatial variance of $s = 2$. Eye localization was performed by correlating the filters over the testing images followed by selecting the peak of the output as the predicted eye location. The eye localization was evaluated by the distance between the predicted and desired eye locations normalized by inter-ocular distance,

$$d = \frac{\|\mathbf{p}_r - \mathbf{m}_r\|_2}{\|\mathbf{m}_l - \mathbf{m}_r\|_2} \tag{16}$$

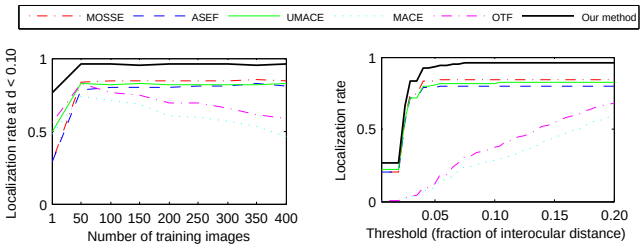
where $\mathbf{m}_r$ and $\mathbf{m}_l$ respectively indicate the true coordinates of the right and left eye, and $\mathbf{p}_r$ is the predicted location of the right eye. A localization with normalized distance $d < th$ was considered as a successful localization. The threshold th was set to a fraction of inter-ocular distance.

The average of evaluation results across 10 random runs are depicted in Figure 2, where our method outperforms the other approaches across all thresholds and training set sizes. The accuracy of OTF and MACE declines by increasing the number of training images due to over-fitting. During the experiment, we observed that the low performance of the UMACE, ASEF and MOSSE was mainly caused by wrong localizations of the left eye and the nose. This was not the case for our method, as our filter was trained in a way that return zero correlation values when centred upon non-target patches of the face image. A visual depiction of the filters and their outputs can be seen in Figure 3, illustrating examples of wrong and correct localizations. The Peak-to-Sidelobe Ratio (PSR) [4] values show that our method returns stronger output compared to the other filters.

Moreover, we examined the influence of T (the size of training images) on the performance of eye localization. For this purpose, we employed cropped patches of the right eye with varying sizes of $T = \{D, 1.5D, 2D, 2.5D, 3D, 3.5D, 4D\}$ to train filters of size $D = 32 \times 32$. The localization results are illustrated in Figure 4(a), showing that the lowest performance obtained when T is equal to D and the localization rate improved by increasing the size of the training patches with respect to the filter size. The reason is that by choosing $T > D$ the portion of patches unaffected by boundary effects $(\frac{T-D+1}{T})$ reduces.

4.2 Runtime Performance

This experiment demonstrates the advantage of our approach to other iterative methods. Specifically, we compared our proposed approach against other methods in literature for learning



(a) (b)

Fig. 2. Eye localization performance as a function of (a) number of training images, and (b) localization thresholds.

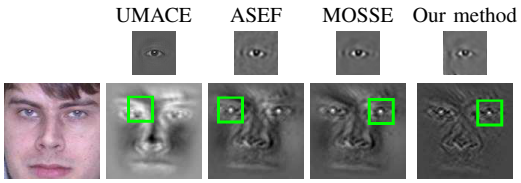


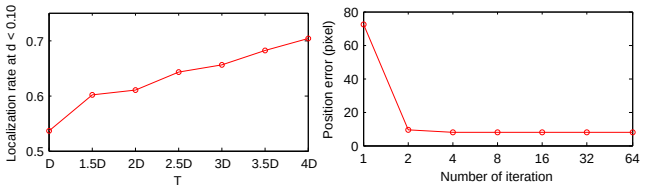
Fig. 3. An example of eye localization is shown for an image with normal lighting. The outputs (bottom) are produced using 64x64 correlation filters (top). The green box represents the approximated location of the right eye (output peak). The peak strength measured by PSR shows the sharpness of the output peak.

filters efficiently using iterative methods. We compared our convergence performance with a steepest descent method [20] for optimizing our same objective. Results can be seen in Figure 5: (a) represents time to converge as a function of the number of training images, and (b) represents the number of iterations required to optimize the objective (in Equation 9).

In (a) one notices impressively how convergence performance is largely independent to the number of images used during training. This can largely be attributed to the precomputation of the auto- and cross-spectral energy vectors. As a result, iterations of the ADMM do not need to re-touch the training set, allowing our proposed approach to dramatically outperform more naive iterative approaches. Similarly, in (b) one also notices how relatively few iterations are required to achieve good convergence.

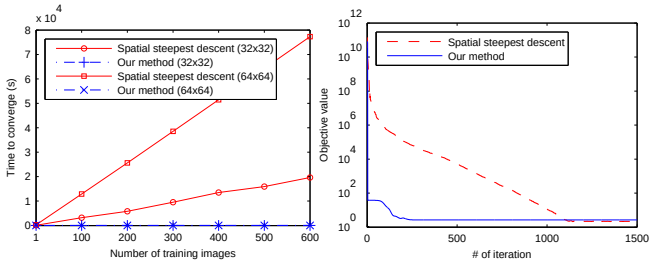
4.3 Tracking Performance

Finally, we evaluated the proposed method for the task of real-time tracking on a sequence of commonly used test videos [18], described in Table 1. We compared our approach with state-of-the-art trackers including MOSSE [4], kernel-MOSSE [11], MILTrack [3], STRUCK [10], OAB [8], SemiBoost [9], FragTrack [1] and IVT [18]. All of these methods were tuned by the parameter settings proposed in their reference papers. The desired response for a $m \times n$ target was defined as a 2D Gaussian with a variance of $\mathbf{s} = \sqrt{mn}/16$. The regularization parameter λ was set to 10^{-2}. We evaluated our method with differen number of iterations $\{1, 2, 4, 8, 16, 32, 64\}$, as shown in Figure 4(b), and eventually



(a) (b)

Fig. 4. (a) The localization rate obtained by different sizes of training images (T), the size of the trained filter is $D = 32 \times 32$. (b) The position error of tracking versus the number of ADMM iterations. We selected 4 iterations as a tradeoff between tracking performance and computation.



(a) (b)

Fig. 5. Runtime performance of our method against another naive iterative method (steepest descent method) [20]. Our approach enjoys superior performance in terms of: (a) convergence speed to train two filters with different sizes (32x32 and 64x64) and (b) the number of iterations required to converge.

selected four iterations (a tradeoff between precision and tracking speed) for our tracker. A track initialization process was employed for our approach and MOSSE, where eight random affine perturbations were used to initialize the first filter. We borrowed the online adaption from the work of Bolme et al. [4] to adapt our filter at i^{th} frame using averaged auto-spectral and cross-spectral energies:

$$\begin{aligned} (\hat{\mathbf{s}}_{xx})^i &= \eta(\hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{x}}_i)) + (1-\eta)(\hat{\mathbf{s}}_{xx})^{i-1} \\ (\hat{\mathbf{s}}_{xy})^i &= \eta(\hat{\mathbf{y}}_i \circ \text{conj}(\hat{\mathbf{x}}_i)) + (1-\eta)(\hat{\mathbf{s}}_{xy})^{i-1} \end{aligned} \quad (17)$$

where, η is the adaption rate. We practically found that $\eta = 0.025$ is appropriate for our method to quickly be adapted against pose change, scale, illumination, etc.

The tracking results are evaluated in Table 2 following the recent tracking papers [3] [10] [8], including (i) percentage of frames where the predicted position is within 20 pixels of the ground truth (precision), (ii) average localization error in pixels, and (iii) tracking speed as frames per second (*fps*) Our method averagely achieved maximum precisions and minimum localization errors, followed by STRUCK. One explanation for this is that our method employs a rich set of training samples containing all possible positive (target) and negative (non-target) patches to train the correlation filter. Whilst, the non filter approaches such as STRUCK and MILTrack are limited by learning a small subset of positive and negative patches.

Sequence	Frames	Main Challenges
Faceocc1	886	Moving camera, occlusion
Faceocc2	812	Appearance change, occlusion
Girl	502	Moving camera, scale change
Sylv	1344	Illumination and pose change
Tiger1	354	Fast motion, pose change
David	462	Moving camera, illumination change
Cliffbar	472	Scale change, motion blur
Coke Can	292	Illumination change, occlusion
Dollar	327	Similar object, appearance change
Twinings	472	Scale and pose change

TABLE 1
Video sequences used for tracking evaluation.

Similarly, it can be explained that the accuracy of MOSSE and kernel-MOSSE are affected by using synthetic negative samples which are not representative of the "real-world" examples, as illustrated in Figure 1(c). Moreover, our method owes its robustness against challenging variations in scale (*Cliffbar* and *Twinings*), illumination (*Sylv*), pose (*David*), appearance (*Girl*) and partial occlusion (*Faceocc1* and *Faceocc2*) to the online adaption. In the case of tracking speed, MOSSE outperformed the other methods by 600 *fps*. Our method obtained lower *fps* compared to MOSSE and kernel-MOSSE, due to its iterative manner. However, it obtained a tracking speed of 50 *fps* which is appropriate for real-time tracking.

A visual depiction of tracking results for some selected videos is shown in Figures 6 and 7, where our method achieved higher precisions over all videos except *Tiger1* and *Twinings*. Moreover, Figure 6(b) shows that our approach suffers from less drift over the selected test videos.

5 Conclusions

A method for estimating a correlation filter is presented here that dramatically limits circular boundary effects while preserving many of the computational advantages of canonical frequency domain correlation filters. Our approach demonstrated superior empirical results for both object detection and real-time tracking compared to current state of the arts.

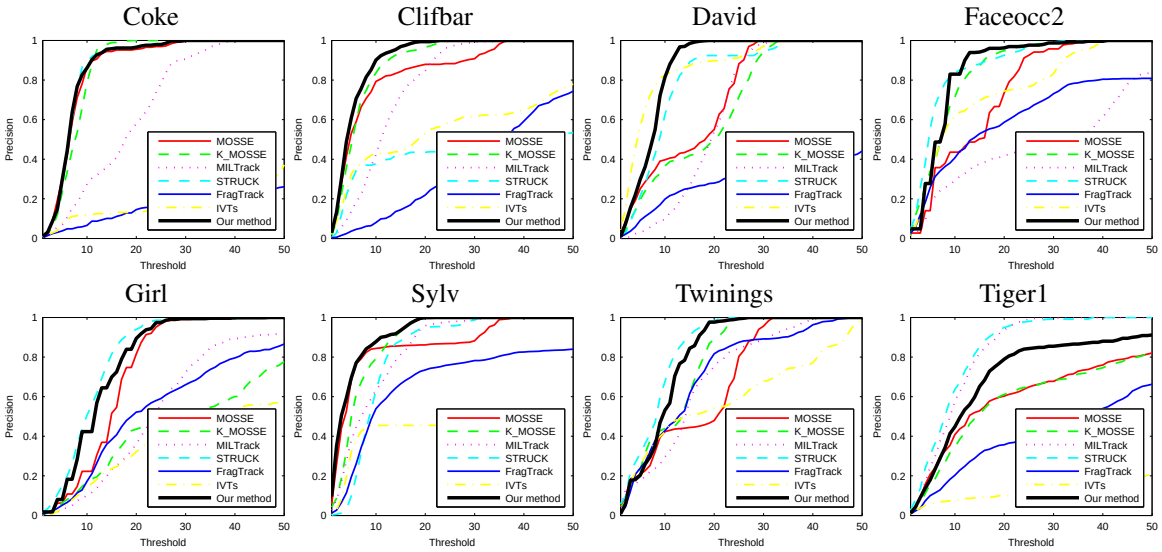
References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments based tracking using the integral histogram. In *CVPR*, 2006.
[2] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.
[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.
[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.
[5] D. S. Bolme, B. A. Draper, and J. R. Beveridge. Average of synthetic exact filters. In *CVPR*, 2009.
[6] S. Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2010.
[7] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini. Bilinear Modelling via Augmented Lagrange Multipliers (BALM). *PAMI*, 34(8):1–14, Dec. 2011.
[8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006.
[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*. Springer, 2008.
[10] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

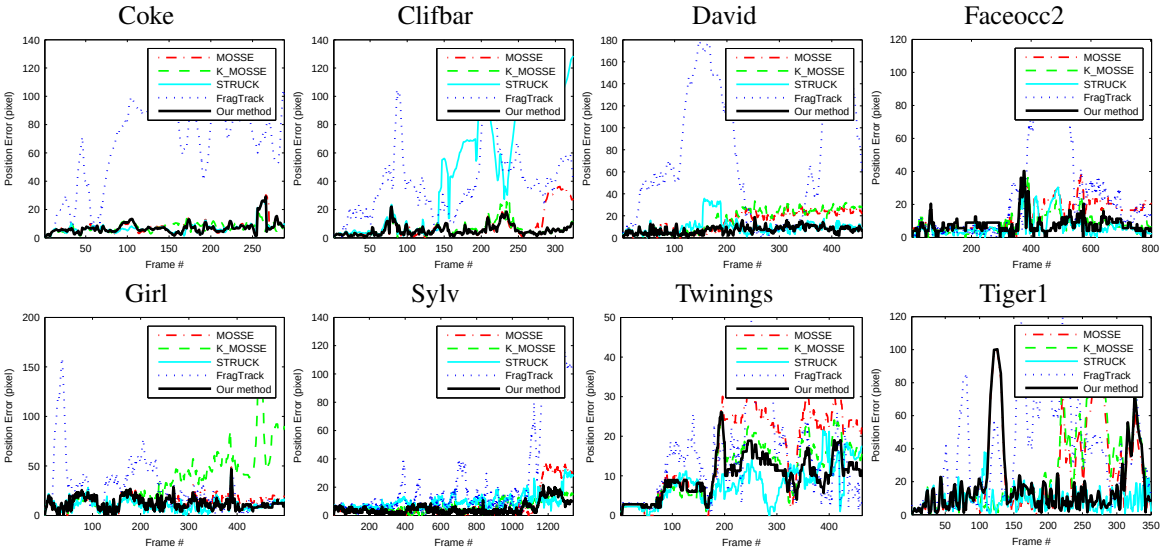
	MOSSE	KMOSSE	MILTrack	STRUCK	OAB(1)	SemiBoost	FragTrack	Our method
FaceOcc1	{**1.00**, 7}	{**1.00**, **5**}	{0.75, 17}	{0.97, 8}	{0.22, 43}	{0.97, 7}	{0.94, 7}	{**1.00**, 8}
FaceOcc2	{0.74, 13}	{0.95, 8}	{0.42, 31}	{0.93, **7**}	{0.61, 21}	{0.60, 23}	{0.59, 27}	{ **0.97**, **7**}
Girl	{0.82, 14}	{0.44, 35}	{0.37, 29}	{**0.94**, **10**}	-	-	{0.53, 27}	{0.90, 12}
Sylv	{0.87, 7}	{**1.00**, 6}	{0.96, 8}	{0.95, 9}	{0.64, 25}	{0.69, 16}	{0.74, 25}	{**1.00**, **4**}
Tiger1	{0.61, 25}	{0.62, 25}	{0.94, 9}	{**0.95**, **9**}	{0.48, 35}	{0.44, 42}	{0.36, 39}	{0.79, 18}
David	{0.56, 14}	{0.50, 16}	{0.54, 18}	{0.93, 9}	{0.16, 49}	{0.46, 39}	{0.28, 72}	{**1.00**, **7**}
Cliffbar	{0.88, 8}	{0.97, 6}	{0.85, 12}	{0.44, 46}	{0.76, -}	-	{0.22, 39}	{ **1.00**, **5**}
Coke Can	{0.96, **7**}	{**1.00**, **7**}	{0.58, 17}	{0.97, **7**}	{0.45, 25}	{0.78, 13}	{0.15, 66}	{0.97, **7**}
Dollar	{**1.00**, **4**}	{**1.00**, **4**}	{**1.00**, 7}	{**1.00**, 13}	{0.67, 25}	{0.37, 67}	{0.40, 55}	{ **1.00**, 6}
Twinings	{0.48, 16}	{0.89, 11}	{0.76, 15}	{**0.99**, **7**}	{0.74, -}	-	{0.82, 14}	{**0.99**, 9}
mean	{0.80, 11}	{0.84, 12}	{0.72, 16}	{0.91, 12}	{0.53, 31}	{0.62, 29}	{0.51, 37}	{**0.97**, **8**}
fps	**600**	100	25	11	25	25	2	50

TABLE 2

The tracking performance is shown as a tuple of {*precision within 20 pixels*, *average position error in pixels*}, where our method achieved the best performance over 8 of 10 videos. The best *fps* was obtained by MOSSE. Our method obtained a real-time tacking speed of 50 *fps* using four iterations of ADMM. The best result for each video is highlighted in bold.



(a)



(b)

Fig. 6. Tracking results for selected videos, (a) precision versus the thresholds, and (b) position error per frame.

Fig. 7. Tracking results of our method over the test videos with challenging variations of pose, scale, illumination and partial occlusion. The blue (dashed) and red boxes respectively represent the ground truth and the positions predicted by our method. For each frame, we illustrate the target, trained filter and correlation output.

[11] J. F. Henriques, R. Caseiro, P. Martines, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[12] C. F. Hester and D. Casasent. Multivariant technique for multiclass pattern recognition. *Appl. Opt.*, 19(11):1758–1761, 1980.

[13] B. V. K. V. Kumar. Minimum-variance synthetic discriminant functions. *J. Opt. Soc. Am. A*, 3(10):1579–1584, 1986.

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

[15] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent. Minimum average correlation energy filters. *Appl. Opt.*, 26(17):3633–3640, 1987.

[16] N. C. Oza. *Online Ensemble Learning*. PhD thesis, U. C. Berkley, 2001.

[17] P. Refregier. Optimal trade-off filters for noise robustness, sharpness of the correlation peak, and horner efficiency. *Optics Letters*, 16:829–832, 1991.

[18] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008.

[19] M. Savvides and B. V. K. V. Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. In *AVSS*, pages 45–52, 2003.

[20] M. Zeiler, D. Krishnan, and G. Taylor. Deconvolutional networks. *CVPR*, 2010.