Target Response Adaptation for Correlation Filter Tracking

Adel Bibi, Matthias Mueller, and Bernard Ghanem

Image and Video Understanding Laboratory (IVUL),
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{adel.bibi,matthias.mueller.2,bernard.ghanem}@kaust.edu.sa

Abstract. Most correlation filter (CF) based trackers utilize the circulant structure of the training data to learn a linear filter that best regresses this data to a hand-crafted target response. These circularly shifted patches are only *approximations* to actual translations in the image, which become unreliable in many realistic tracking scenarios including fast motion, occlusion, etc. In these cases, the traditional use of a single centered Gaussian as the target response impedes tracker performance and can lead to unrecoverable drift. To circumvent this major drawback, we propose a generic framework that can adaptively change the target response from frame to frame, so that the tracker is less sensitive to the cases where circular shifts do not reliably approximate translations. To do that, we reformulate the underlying optimization to solve for both the filter and target response *jointly*, where the latter is regularized by measurements made using actual translations. This joint problem has a closed form solution and thus allows for multiple templates, kernels, and multi-dimensional features. Extensive experiments on the popular OTB100 benchmark [18] show that our target adaptive framework can be combined with many CF trackers to realize significant overall performance improvement (ranging from 3%-13.5% in precision and 3.2%-13% in accuracy), especially in categories where this adaptation is necessary (e.g. fast motion, motion blur, etc.).

Keywords: Correlation Filter Tracking, Adaptive Target Design

1 Introduction

Visual object tracking is a classical problem in computer vision. It plays an important role in a plethora of applications, such as robotics, surveillance, and human-computer interaction to name a few. Object tracking can be defined as the task of localizing an object of interest (e.g. by an upright bounding box) in every frame starting from a given patch containing the object in the first frame. The problem is very challenging because the object could undergo a variety of transformations making it harder to localize. Typical nuisances that have to be overcome by a successful object tracker include occlusion, in- and out-of-plane rotation, fast motion, illumination changes, etc.

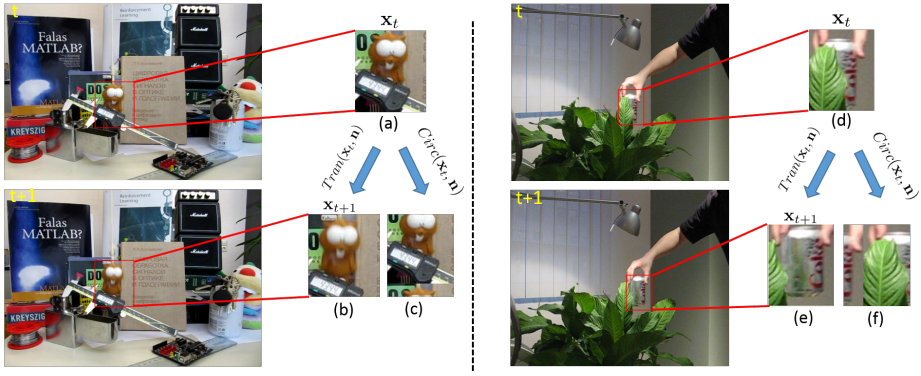

Fig. 1. Shows examples where circular shifts do not represent actual translations. Patches (a) and (b) of video *Lemming* show the object in two consecutive frames, where the target was partially occluded and the occluder is within the filter window. The circular shift corresponding to the actual translation of the object in the next frame is given in patch (c). Note that both the occluder and target are shifted. $\text{Circ}(\mathbf{x}, \mathbf{n})$, and $\text{Tran}(\mathbf{x}, \mathbf{n})$ denote $\mathbf{n}$ circular shifts and actual translations applied to the patch $\mathbf{x}$, respectively. Similarly, we show patches (d) and (e) of video *Coke* of two consecutive frames, where fast motion and partial occlusion occur. The corresponding circular shift is given in patch (f). In both examples, translations and their approximations (circular shifts) are quite different. This discrepancy will severely affect the detection step (and in turn the training step) of any CF based tracker at that frame.

CF based trackers [5, 10, 11, 8, 13] have gained much attention lately for their attractive performance both in speed and accuracy. The key idea behind CF trackers is that a learned filter is used to localize the object in the next frame by identifying the location of maximal correlation/convlution response (detection step). Then, it is updated by computing a filter, whose correlation with training templates (most often the current tracking result) closely resembles a hand-crafted target response, usually taken to be a Gaussian centered at the current tracking result (training step) [10, 11, 8, 13, 9, 12]. A recent development in this tracking paradigm and the main reason behind its computational efficiency is the use of circulant structure in the training step. In many cases (e.g. when the background is homogenous and no occlusion occurs), the circular shifts of the training templates actually represent translations in the image domain. This means that the *motion* of a template is inherently accounted for by these circular shifts.

Despite its merits, there are two main drawbacks in the traditional CF tracking paradigm. **(i)** Since the detection step of the tracker might be inaccurate (e.g. due to fast motion, motion blur, etc.), the localization of the object in the next frame is erroneous. Moreover, since the target response is independent of the frame, error will be propagated to the newly computed filter and the tracker becomes at risk of unrecoverable drift. **(ii)** The target response used in the training step is independent of the observed frame and assumes that circular

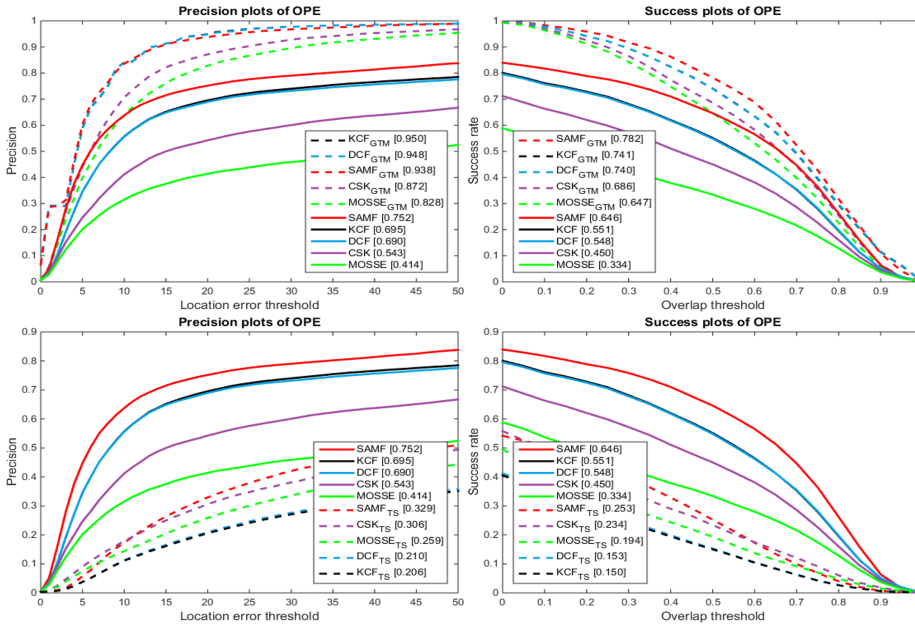

Fig. 2. The first row demonstrates the impressive performance of five CF based trackers (MOSSE_{GTT}, DCF_{GTT}, CSK_{GTT}, KCF_{GTT}, and SAMF_{GTT}) when the target response is obtained from the ground truth of OTB100 [18]. The second row shows the sensitivity of the tracking results to the target response. By only perturbing the target response by at most 2 pixels, the performance drops significantly. This motivates the importance of designing more robust and effective target response.

shifts correspond to actual translations, which is not the case in some scenarios (refer to Figure 1). Obviously, this approximation is not reliable for many tracking nuisances including fast motion, occlusion, and motion blur. Since the target response is not adaptive to the observed frame, the tracker cannot easily recover from errors in the detection step. In this paper, we propose to tackle both drawbacks by jointly solving for the best filter and target response in each frame, where the latter is regularized using actual translation measurements of correlation and not approximated using circular shifts.

As mentioned earlier, the selection of the target response in CF tracking is intimately related to the assumed motion model. Traditionally, this model is simplistic, strict, and prone to drift. Therefore, using/designing a more effective motion model (or equivalently, designing a better target response) is crucial. We justify this observation in Figure 2 by changing the traditional target response in two different ways and reporting the precision and accuracy results of several CF based trackers in both cases. Note that the traditional detection step is not altered in either case.

In the first experiment (top row of Figure 2), the target response (motion model) is *optimal* because it is generated by centering the traditional Gaussian

target at the ground truth object location in each frame, irrespective of the detection location. As compared to using the traditional response, all CF trackers perform significantly better (in both metrics), especially those that use simple grayscale features, which usually make it difficult to reliably detect the object in the next frame. For example, the precision of the basic MOSSE tracker [5] increases from 14% to 82%. Note that perfect performance is not achieved here (even for trackers with scale adaptation) because the detection step still introduces errors. This experiment suggests that it is useful to design more realistic target responses, instead of only focusing on incorporating more complex features that tend only to impact the detection step in CF tracking. In the second experiment (bottom row of Figure 2), perturbations in the detection step are simulated by randomly shifting the traditional Gaussian target response around the detected location by at most 2 pixels in each frame. Clearly, CF tracking performance drops significantly for all methods because it could not recover from the drift. We also conclude from this experiment that designing a better target response is important for more robust CF tracking. This design should be able to handle errors/perturbations in the detection step, thus, allowing the tracker to recover from drift.

2 Related Work

There have been many advances in the field of object tracking, so we only provide a brief overview of the two main categories of trackers (generative and discriminative) in the literature and focus on those that are most relevant to our proposed framework (CF trackers).

Generative Trackers. They adopt an appearance model to describe a set of target observations. The aim of these trackers is to search for the target that is best represented by the updated generative model. Therefore, learning a representative appearance model that can identify the target, even when it undergoes appearance changes is the main emphasis of these trackers. Examples of generative trackers include incremental tracker (IVT) [17], mean shift tracker [6], L1-min tracker [16], multi-task tracker (MTT) [22], low-rank sparse tracker [21], and structural sparse tracker [23] to name a few.

Discriminative Trackers. They formulate object tracking as a classification problem, where the regions around the previous location of the target are given scores to regress to (e.g. using a classifier to predict background or foreground). Examples of discriminative trackers include multiple instance learning (MIL) [3], ensemble tracking [1], support vector tracking [2], and correlation filter (CF) based trackers like [5, 10, 11, 14].

CF Trackers. Using correlation filters for tracking started with Bolme *et al.* [5], where the formulation was constructed in the frequency domain for efficiency, thus, reaching a runtime of 600-700 FPS. Seminal followup work by Henriques *et al.* [10, 11] formulate the problem in the spatial domain, but solved it efficiently in the frequency domain. This is possible by exploiting circulant structure in

the optimization. This method (denoted as the kernel correlation filter tracker or KCF [11]) can incorporate both non-linear kernels (e.g. Gaussian) and multi dimensional features (e.g. HOG). Many improvements have recently been made to this popular tracker to address several limiting issues. For instance, the work in [14, 7] proposes an adaptive scale version of KCF and also made use of the colornames feature. Another approach proposes a multi-template version of KCF [4] by solving a constrained ridge regression problem. Recent work extends KCF to enable part-based tracking [15], where multiple KCF trackers (one for each part) are run independently and the response map of all trackers is computed.

Another variant of CF based trackers changes the objective to spatially focus the filter energy at the center, thus, reducing undesirable boundary effects [8]. Unlike other CF based trackers, their final formulation cannot fully exploit circulant matrix structure into having a closed form element wise solution. Similarly, Galoogahi *et al.* [13] propose a method to deal with the boundary effects of circularly shifted patches by pre-multiplying them with a masking matrix; however, the resulting optimization is also unable to exploit circular structure.

Contributions. To the best of our knowledge, we are the first to investigate the effect of designing an adaptive target response in the context of CF tracking. **(2)** Unlike previous work that fixes the target response for all frames, our proposed method adaptively changes it to account for motion and boundary effects caused by circular shifts. The resulting joint optimization can be solved efficiently by exploiting the underlying circulant structure. **(3)** Extensive experiments on the popular online tracking benchmark OTB100 [18] show that our adaptive target framework can be applied to many CF trackers to improve their performance, while remaining computationally efficient.

3 Correlation Filter Tracking

Similar to other discriminative methods, correlation filters need a set of training examples to learn a filter. In tracking, the first image patch is the only available training example. Other discriminative trackers usually collect positive examples from image patches close to the object in the first frame and negative ones from patches that are farther away. Computational complexity can increase significantly as the number of training patches increases. However, CF based trackers collect dense samples by circularly shifting the patch in the first frame (thus approximating translations) to construct a circulant matrix, which has very desirable properties.

Assuming for simplicity that all the training examples are 1D where $\mathbf{x} \in \mathbb{R}^n$ represents the template in the first frame. Then, $\mathbf{P}\mathbf{x} = [x_n, x_1, \ldots, x_{n-1}]$ represents 1 circular shift of $\mathbf{x}$, where $\mathbf{P}$ is a permutation matrix. Concatenating the set of all possible circularly shifted templates forms a circulant matrix that is also referred to as the data matrix. Correlation filtering seeks a filter $\mathbf{w}$ that minimizes the following ridge regression problem:

$$\underset{\mathbf{w}}{\text{minimize}} \;\; ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 + \lambda||\mathbf{w}||_2^2 \tag{1}$$

The data matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$ can either contain the template $\mathbf{x}$ and all its shifts or a kernelized version of them when using circulant structure preserving kernels [11]. Here, $\mathbf{y}$ is the target response, which is usually assumed to be Gaussian centered around the base patch. Eq 1 can be solved in either the primal or the dual domain each with its own pros and cons.

Primal Domain. Here, the filter $\mathbf{w}$ (the primal variable) is computed to solve Eq 1, which admits a closed form solution in the primal domain given by $\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$. Due to the circulant structure of $\mathbf{X}$, it can be diagonalized and the matrix inversion can be done efficiently. The filter solution (in FFT form) is given by $\hat{\mathbf{w}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}$, where $\hat{\mathbf{w}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{x}}$ are the FFTs of $\mathbf{w}$, $\mathbf{y}$, and $\mathbf{x}$, respectively. The $*$ denotes the complex conjugate and all operations are element wise [11]. The primal formulation also enables the use of multiple templates without changing the solution mechanism much. In this case, Eq 1 is solved with $\mathbf{X}$ replaced with $\tilde{\mathbf{X}}$, which is the blockwise concatenation of the circulant matrices of all the templates. It can be easily shown [11] that the optimal filter for k templates is given by $\hat{\mathbf{w}} = \frac{\sum_{j=1}^{k} \hat{\mathbf{x}}_j^* \odot \hat{\mathbf{y}}}{\sum_{j=1}^{k} \hat{\mathbf{x}}_j^* \odot \hat{\mathbf{x}}_j + \lambda}$. However, this formulation does not facilitate the use of kernels because the solution is not written as a function of bi-products of the circulant matrices.

Dual Domain. Conversely, Eq 1 can also be solved in the dual domain, where the solution is $\alpha = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{y}$. Here, α is the dual variable and it is related to the primal variable through $\mathbf{w} = \mathbf{X}^T\alpha$. The dual formulation admits the solution as a function of bi-products of the circulant data matrix allowing the use of the kernel trick. The dual solution (in FFT form) is $\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}$, where $\hat{\alpha}$ is the FFT of α. Unfortunately, using k templates in the dual domain can no longer be done efficiently because $\mathbf{X}\mathbf{X}^T$ is now a matrix with circulant blocks, which has an inversion computational complexity of $n^2O(k^3)$ compared to $kO(nlog(n))$ in the primal domain.

4 Learning Adaptive Target Responses for CF Tracking

As seen in Figure 2, exploiting a reliable motion model in CF based tracking (i.e. designing a better target response $\mathbf{y}$) can significantly boost performance. In this paper, we do this by adaptively changing $\mathbf{y}$ in every frame. The peak values of $\mathbf{y}$ not only favor a training template over another based on appearance information, but also based on prior motion information as well. To do this, we solve the following joint optimization:

$$\underset{\mathbf{w},\mathbf{y}}{\text{minimize}} \;\; ||\tilde{\mathbf{X}}\mathbf{w} - \mathbf{y}||_2^2 + \lambda_1||\mathbf{w}||_2^2 + \lambda_2||\mathbf{y} - \mathbf{y}_o||_2^2 \tag{2}$$

As compared to the classical CF formulation in Eq 1, ours does not assume that the target response $\mathbf{y}$ is known apriori, but instead its constructed prior $\mathbf{y}_o$

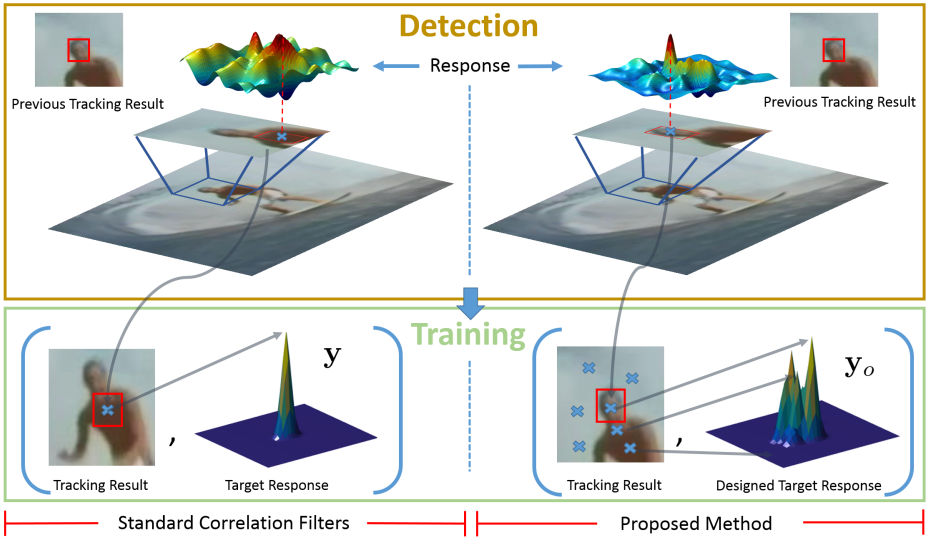

Fig. 3. Shows the pipeline of both standard CF based trackers and our approach. Both involve a detection and training step with a key difference that our approach uses the current detection frame to sample actual translations, which will be used to construct a prior for the target response. In fact, standard CF tracking is a special case of our formulation. When only one translation is sampled around the current tracking result, our approach reduces to the standard CF model.

is known. In what follows, we discuss how $\mathbf{y}_o$ is computed, how Eq 2 is solved for k templates with $\hat{\mathbf{X}} \in \mathbb{R}^{kn \times n}$ being a block circulant matrix, and how the single template solution follows directly. We will also provide the new detection equation for the objective in Eq 2, as well as, an exposition of how our method differs from all other CF based trackers (refer to Figure 3). All the derivations are done for a 1D example, but they can be easily extended to 2D. More details can be found in the **supplementary material**.

Construction of $\mathbf{y}_o$. In Eq 2, the target $\mathbf{y}$ is assumed to follow the noise model: $\mathbf{y} = \mathbf{y}_o + \mathbf{n}$, where $\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{y}_o, \text{diag}^{-1}(\frac{1}{2\lambda_2}))$. At the first frame, a standard KCF [11] filter is learnt by solving Eq 1. In the next frame, a fixed number p translations are sampled, at which the previous filter is correlated with the image. These correlations are used to fill the corresponding p entries in $\mathbf{y}_o$. As motivated earlier, this process generates correlation scores for actual translations, which can offset the limiting effects of using their approximations (i.e. circular shifts). We choose $p \ll n$, so the computational burden remains reasonable for our tracking scenario. The rest of the entries in $\mathbf{y}_o$ are computed from the p computed values by Gaussian interpolation. We did experiment with more sophisticated types of interpolation (e.g. bilateral filtering using the image patch as a guide). This resulted in unnoticeable change in performance and an increase in computation cost.

In subsequent frames and to encode motion information, the aforementioned translations are sampled using a standard Kalman filter with a constant velocity motion model. Other motion models can be used here, such as particle filters.

Multiple Template Solution to Eq 2. Here, $\tilde{\mathbf{X}}^\top = \begin{bmatrix}\mathbf{X}_1^\top & \mathbf{X}_2^\top & \dots & \mathbf{X}_k^\top\end{bmatrix} \in \mathbb{R}^{kn\times n}$, which is the concatenation of all the circulant matrices generated from all the templates. By introducing the variable $\mathbf{z}^\top = \begin{bmatrix}\mathbf{w}^\top & \mathbf{y}^\top\end{bmatrix}$, Eq 2 (in its primal form) can be written as:

$$\underset{\mathbf{z}}{\text{minimize}} \;\; \|\tilde{\mathbf{G}}\mathbf{z}\|_2^2 + \lambda_1\|\mathbf{E}\mathbf{z}\|_2^2 + \lambda_2\|\mathbf{D}\mathbf{z} - \mathbf{y}_o\|_2^2, \tag{3}$$

where $\tilde{\mathbf{G}} = \begin{bmatrix}\tilde{\mathbf{X}} & -\tilde{\mathbf{I}}\end{bmatrix} \in \mathbb{R}^{kn\times 2n}$, $\tilde{\mathbf{I}}^\top = [\mathbf{I} \;\cdots\; \mathbf{I}]$, $\mathbf{E} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{n\times 2n}$, and $\mathbf{D} = \begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix} \in \mathbb{R}^{n\times 2n}$. The problem is convex quadratic, so a global solution can be easily derived (refer to **supplementary material** for details of this derivation). In its dual form, Eq 3 becomes:

$$\underset{\alpha}{\text{minimize}} \; \|\mathbf{D}\tilde{\mathbf{K}}^{-1}\mathbf{D}^T\alpha - \mathbf{y}_o\|_2^2 + \lambda_1\|\mathbf{E}\tilde{\mathbf{K}}^{-1}\mathbf{D}^T\alpha\|_2^2 + \|\mathbf{G}\tilde{\mathbf{K}}^{-1}\mathbf{D}^T\alpha\|_2^2, \tag{4}$$

where α is the dual variable and is related to $\mathbf{z}$ through $\mathbf{z} = \tilde{\mathbf{K}}^{-1}\mathbf{D}^T\alpha$ and $\tilde{\mathbf{K}} = \Big(\lambda_1\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\mathbf{G}\Big)$. Solving Eq 4 is straightforward, as it is equivalent to solving the following linear system:

$$\mathbf{D}\tilde{\mathbf{K}}^{-1}\Big(\lambda_2\mathbf{D}^T\mathbf{D} + \lambda_1\mathbf{E}^T\mathbf{E} + \mathbf{G}^T\mathbf{G}\Big)\tilde{\mathbf{K}}^{-1}\mathbf{D}^T\alpha = \lambda_2\mathbf{D}\tilde{\mathbf{K}}^{-1}\mathbf{D}^T\mathbf{y}_o \tag{5}$$

By using the inverse lemma, the closed form solution to Eq 5 is:

$$\hat{\alpha}^* = \lambda_2 diag^{-1}(\Upsilon)\left(\frac{\frac{1}{k}\Big(\sum_i^k \hat{\mathbf{x}}_{1i}^*\Big)\odot\Big(\sum_i^k \hat{\mathbf{x}}_{1i}\Big)\odot\hat{\mathbf{y}}_0^*}{\sum_i^k(\hat{\mathbf{x}}_{1i}^*\odot\hat{\mathbf{x}}_{1i}) + \lambda_1 - \frac{1}{k}(\sum_i^k\hat{\mathbf{x}}_{1i}^*\odot\sum_i^k\hat{\mathbf{x}}_{1i})} + \frac{\hat{\mathbf{y}}_o^*}{k}\right), \tag{6}$$

where

$$\Upsilon = \left(\frac{\frac{-1}{k}\sum_i^k(\hat{\mathbf{x}}_{1i}^*\odot\hat{\mathbf{x}}_{1i}) + \frac{k+\lambda_2}{k}(\sum_i^k\hat{\mathbf{x}}_{1i}^*)\odot(\sum_i^k\hat{\mathbf{x}}_{1i}) + \frac{\lambda_1(k+\lambda_2)}{k}}{\sum_i^k(\hat{\mathbf{x}}_{1i}^*\odot\hat{\mathbf{x}}_{1i}) + \lambda_1 - \frac{1}{k}(\sum_i^k\hat{\mathbf{x}}_{1i}^*)\odot(\sum_i^k\hat{\mathbf{x}}_{1i})}\right)\odot \left(\frac{\frac{1}{k^2}\sum_i^k\hat{\mathbf{x}}_{1i}^*\odot\sum_i^k\hat{\mathbf{x}}_{1i}}{\sum_i^k(\hat{\mathbf{x}}_{1i}^*\odot\hat{\mathbf{x}}_{1i}) + \lambda_1 - \frac{1}{k}(\sum_i^k\hat{\mathbf{x}}_{1i}^*)\odot(\sum_i^k\hat{\mathbf{x}}_{1i})} + \frac{1}{k}\right) \tag{7}$$

Here, $\hat{\mathbf{x}}_{1i}$ denotes the FFT of the first row of $\mathbf{X}_i$ where all the operations are element wise and thus computationally attractive. Moreover, the solution for one single template can be easily found by setting $k = 1$ in Eq 6 to obtain:

$$\hat{\alpha} = \frac{\Big(\frac{\lambda_2}{\lambda_1}(\hat{\mathbf{x}}_1\odot\hat{\mathbf{x}}_1^*) + \lambda_2\Big)\odot\hat{\mathbf{y}}_o}{\frac{\lambda_2}{\lambda_1^2}(\hat{\mathbf{x}}_1\odot\hat{\mathbf{x}}_1^*\odot\hat{\mathbf{x}}_1\odot\hat{\mathbf{x}}_1^*) + \frac{1+2\lambda_2}{\lambda_1}(\hat{\mathbf{x}}_1\odot\hat{\mathbf{x}}_1^*) + (1+\lambda_2)}, \quad \text{for} \;\; k = 1 \tag{8}$$

Detection Formula. The previous solution is used to train the filter $\mathbf{w}$ or the dual variables α. As for the detection, a similar approach is used as in [11],

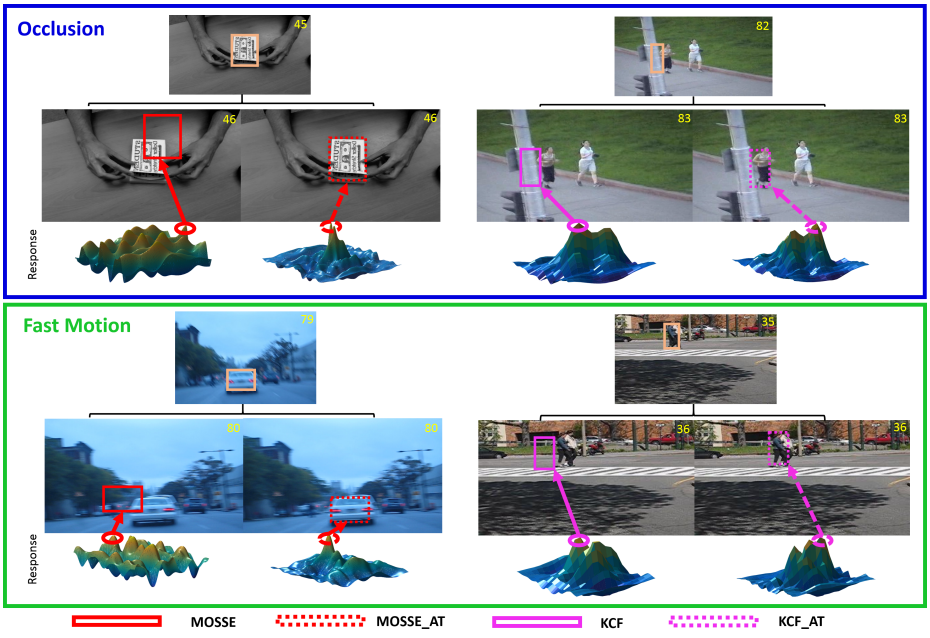

Fig. 4. The first row shows tracking results on occlusion sequences (from left to right: *Coupon* and *Jogging1*) for MOSSE and KCF, along with their adaptive target versions MOSSE$_{AT}$ and KCF$_{AT}$. In the second row, we show similar results for two fast motion sequences (from left to right: *BlurCar2* and *Couple*) for the same trackers.

where a circulant data matrix of the test sample $\mathbf{u}$ is considered for detection. The following is the detection formula for a single template case:

$$\mathbf{T}(\mathbf{u}) = \mathbf{U}\mathbf{w} = \mathbf{X}^T\alpha = \frac{1}{\lambda_1}\mathbf{F}diag(\hat{\mathbf{u}} \odot \hat{\mathbf{x}}_1^*)\hat{\alpha}^* \;\Rightarrow\; \hat{\mathbf{T}}(\mathbf{u}) \;= \frac{1}{\lambda_1}\hat{\mathbf{u}}^* \odot \hat{\mathbf{x}}_1 \odot \hat{\alpha}, \quad (9)$$

where $\hat{\mathbf{T}}$ is the FFT of the detection over all circular shifts of a sample $\mathbf{u}$. It is important to note that when $\lambda_2 \to \infty$ the soft constraint becomes a hard one, where our formulation reduces back to the original CF tracking formulation with a target response $\mathbf{y}_o$. Therefore, the standard CF tracking framework can be viewed as a special case of our adaptive formulation.

Comparison to CF Based Trackers. As discussed earlier, CF based trackers [5, 10, 11, 8, 14], as shown in Figure 3, exploit two main steps: detection and training while the target response used during the training is assumed to be independent of the frame and taken to be Gaussian centered at the window center. This inherently assumes that the detected location of the window is correct.

When errors (even as small as a few pixels) arise in the detection, the target response $\mathbf{y}$ is not centered properly and these errors propagate into the filter estimation. This error propagation usually leads to tracker drift if multiple

subsequent detection errors are encountered. This is illustrated in Figure 2. Obviously, this detection/training process is not fault tolerant and has difficulties recovering from errors. In comparison, our approach assumes that $\mathbf{y}$ is unknown and estimates it at every frame by making use of a target response prior $\mathbf{y}_o$, which exploits correlation values at actual translations in the next frame to help the filter update regress to more realistic target values. As such, our proposed strategy is less prone to error propagation than the classical CF procedure. We illustrate this conclusion with a qualitative example in Figure 4, where two trackers (MOSSE and KCF) are compared against their target adpative versions, when they encounter occlusion and fast motion. When our adaptive target method is used, the corresponding response maps are less noisy with the simpler tracker (MOSSE) and better localized with KCF (that uses more sophisticated features). The response map is biased towards the correct target location, since actual translations are used in the correlation measurements of the training step.

5 Experiments

We validate our adaptive target response framework by integrating it into five popular CF-based trackers. The experiments are run on the OTB100 dataset [18], which comprises 100 challenging video sequences including all 50 videos from its previous version OTB50 [19]. As compared to other tracking datasets, OTB100 contains a higher percentage of sequences that experience fast motion, motion blur, and occlusion.

Baseline Trackers. They differ in terms of the features used, kernels applied, and their ability to adapt to object scale variations. Particularly, MOSSE [5] uses grayscale features and a linear kernel, while CSK [10] uses the same features but with a gaussian kernel. DCF [11] uses HOG features along with a linear kernel, while KCF [11] uses the same features but with a gaussian kernel. The four aforementioned trackers do not adapt to scale changes, so we choose SAMF [14] to represent CF-based trackers that are scale variant. Note that DSST [7] is another option of this type, but we only include SAMF in our evaluation because it outperforms DSST on OTB100 [18] and their methodology is very similar. Applying our framework to the five baseline trackers gives rise to their adaptive target variants: MOSSE_{AT}, CSK_{AT}, DCF_{AT}, KCF_{AT}, and SAMF_{AT}.

In fact, our framework can be applied to any CF-based tracker, but the aforementioned ones (and most trackers of this type in general) use a formulation that allows for the direct and efficient implementation of our target adaptation. Other trackers, such as SRDCF [8], are included in the evaluation but not modified for target response adaptation because the closed form solutions in Eqs 6&8 do not apply directly to the underlying optimization in these trackers. For example, SRDCF adds spatial regularization to $\mathbf{w}$, which impedes the effective exploitation of circulant structure. Nevertheless, we provide details in the **supplementary material** on how our framework can be extended to include SRDCF and trackers with similar formulations.

Implementation Details and Parameters. In all our experiments, we use MATLAB on an Intel(R) Xeon(R) 2.67GHz CPU with 32GB RAM. For all the baseline trackers, we use the original parameters provided by the authors. The best regularization parameters λ_1 and λ_2 are selected for each baseline tracker. They are $\{(10^{-1}, 10^{-2}), (10^{-1}, 10^{-4}), (10^{-6}, 10^{-2}), ((10^{-3}, 10^{-5}), ((10^{-3}, 10^{-2})\}$ for MOSSE$_{AT}$, CSK$_{AT}$, DCF$_{AT}$, KCF$_{AT}$, and SAMF$_{AT}$ respectively. For simplicity and fair comparison, we consider $k = 1$ templates in our experiments for all trackers. The standard update rule for the newly computed filter [5, 10, 11] is used, where the learning rate is set to $(0.02, 0.01, 0.01, 0.01, 0.015)$ for MOSSE$_{AT}$, CSK$_{AT}$, DCF$_{AT}$, KCF$_{AT}$, and SAMF$_{AT}$ respectively.

As for the number of translations used to form the prior target response $\mathbf{y}_o$, we set $p = 13$ for trackers with grayscale features (MOSSE$_{AT}$ and CSK$_{AT}$) and $p = 7$ for those with HOG features (DCF$_{AT}$, KCF$_{AT}$, and SAMF$_{AT}$). This discrepancy is due to cell size used in HOG (or any other patch based feature). The granularity of each translation is dependent on the cell size of the feature, which is taken to be 4 pixels for HOG. In this case, the minimum translation possible would be 4 pixels, thus, allowing for larger translations with a smaller number of translation samples. For the case of $p = 13$, translations are initialized from the set of $\{0, 3, -3, 5, -5\}$ pixels in a grid fashion, while this set is $\{0, 4\}$ when $p = 7$. Several expirements have been conducted on several choices of p, but it turns out that increasing p beyond 13 have marginal impact on performance. The padding region was set to 2 for all trackers. Moreover, the scaling function of SAMF$_{AT}$ is the same as that of SAMF, where 9 scales are considered with the same step size as the original implementation [14].

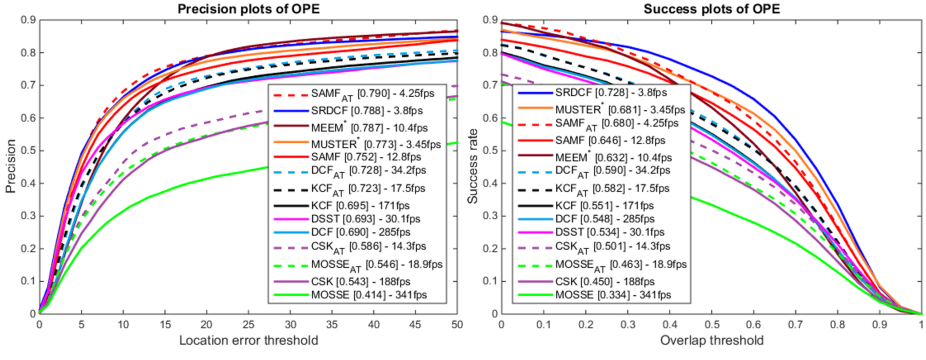

Fig. 5. Precision and accuracy results for five baseline CF trackers, their adaptive target variants, as well as, other state-of-the-art methods. Trackers denoted by * are either not CF trackers or only use a CF tracker as a baseline for a generic framework.

Quantitative Results. We run all baseline and adaptive target trackers on OTB100 [18]. Following its standard evaluation strategy, we show the overall precision and accuracy plots of all trackers in Figure 5. The precision is defined

as the average number of frames per video that are at most 20 pixels away from the ground truth, while the accuracy is defined as the average number of frames per video where the intersection over union with ground truth is at least 0.5. For a complete comparison, we also show the results of other state-of-the-art trackers including, SRDCF [8], MUSTER [12], MEEM [20], and DSST [7]. All trackers with target adaptation improve in performance, where the improvement ranges from 4% for sophisticated trackers like SAMF to 15% for basic ones like MOSSE. It is worthwhile to note that SAMF$_{AT}$ achieves state-of-art performance in precision and is tied with MUSTER for second place right after SRDCF in accuracy. The reason behind this ranking discrepancy between the two metrics is primarily due to the scaling modality used in SAMF. Evidence of this phenomenon also arises in Figure 2, where SAMF accuracy is worse than its precision, even when the target response is optimal.

In Figure 6, we show an extensive comparison of the baseline trackers and their adaptive target variants on sequences with attributes (fast motion, motion blur, occlusion, and low resolution) that are expected to benefit the most from our adaptive framework. In fact, the performance of all the baseline trackers improves with target adaptation, some more than others in certain attributes. In general, trackers that use multi-dimensional sophisticated features experience less improvement than those that use grayscale features. For example, since there are more severe object translations in the subset of videos in the motion blur category that do not belong to the fast motion category, the range of improvement in the former category (6%-24.1%) is higher than the latter (2.6%-25.8%). In the occlusion category, the trackers with grayscale features (MOSSE and CSK) are the only ones with significant improvement (13% and 7.7% respectively). On the other hand, trackers that use sophisticated features and/or non-linear kernels have less improvement (i.e. DCF$_{AT}$, KCF$_{AT}$, and SAMF$_{AT}$). The reason behind this non-uniform improvement among trackers is two-fold. First, the occlusion category comprises about 50% of the whole OTB dataset, so occlusion videos contain many other attributes (some that do not benefit from target adaptation), thus, making the improvement less obvious. Secondly, more sophisticated features (e.g. HOG) play an important role in making the detection step of the CF tracker more robust to occlusion. The low resolution category witnesses the largest improvements overall. Interestingly, for videos with this attribute, basic trackers like MOSSE and CSK can outperform more established trackers like SAMF, when they exploit target adaptation. In fact, even SAMF improves by 15.7% here. Since our method makes use of correlation scores from actual translations to bias the target response, the learned filter is better at localizing a smaller object (i.e. whose dimensions are more comparable to its frame-to-frame translation), when compared to traditional CF trackers that only use circular shifts. This is because a standard cosine window is applied to the patch [10, 11] which is proportional to object's size. This limits the motion search for the object in standard CF trackers unlike our method that allows for the detection of larger translations.

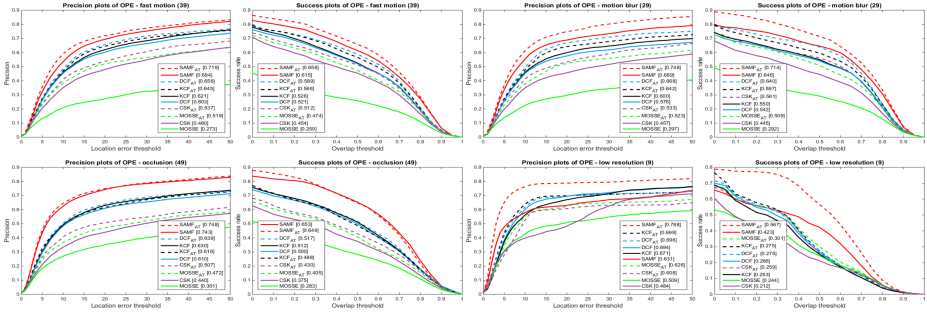

Fig. 6. Precision and accuracy results for the fast motion, motion blur, occlusion, and low resolution categories in OTB100 [18].

Qualitative Results. Figure 7 shows qualitative results comparing the five baseline trackers to their target adaptive variants. For the first row (the *BlurOwl* sequence), the target undergoes fast motion along with motion blur. Unlike SAMF, SAMF$_{AT}$ is able to track the object throughout the complete sequence. In the second row (the *Human4* sequence), KCF is unable to keep tracking the target as it undergoes partial occlusion. When the occluder appears inside the filter window, the circular shifts are no longer good approximations of actual translations and the tracker drifts. Similar behaviour arises when the DCF, KCF, and MOSSE trackers are applied to the *Freeman4*, *Coke*, and *Woman* sequences, respectively. In all these sequences, the target adaptive version of each CF tracker is able to consistently maintain the track.

6 Conclusions

In this paper, we propose a generic framework for correlation filter (CF) based trackers to counter the problem of fast motion, motion blur, and occlusion in videos. Our approach efficiently solves for both the filter and target response jointly, whereby the target response is regularized using correlation scores evaluated at sampled translations. Experiments demonstrate significant improvement in performance when our adaptive target framework is applied to many CF trackers. The proposed method is generic and can be incorporated into any CF based tracker. For future work, we aim to investigate more systematic and effective strategies for sampling the translations from frame to frame.

Acknowledgments. Research in this publication was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research.

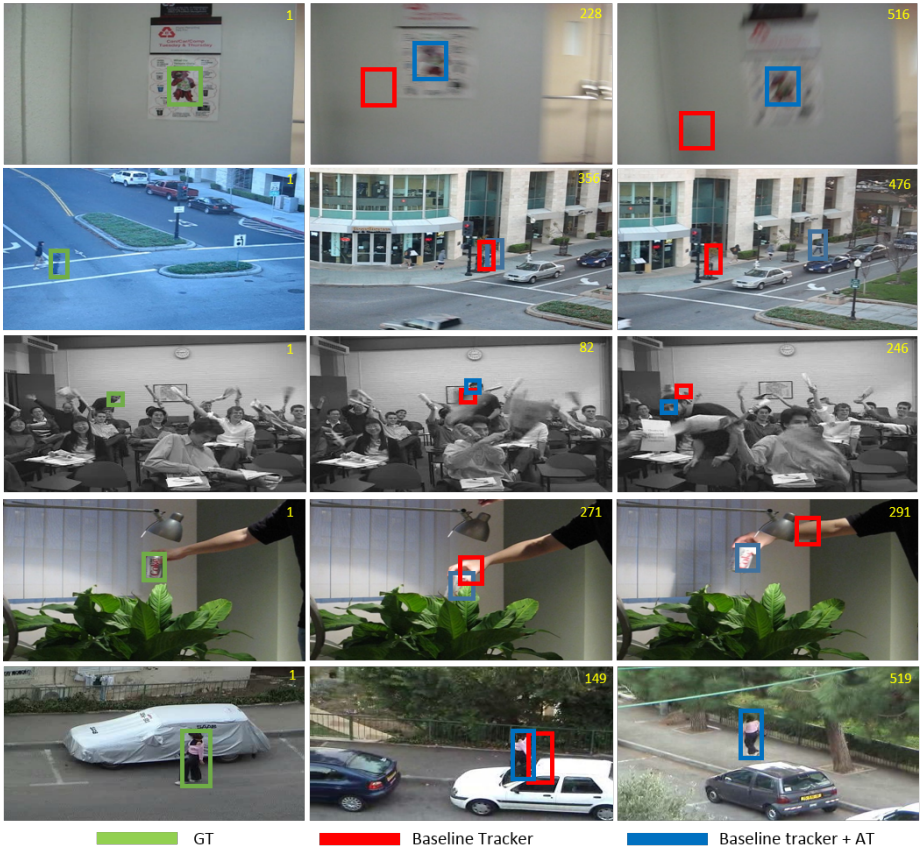

Fig. 7. Shows tracking results for five videos (from top to bottom: *BlurOwl*, *Human4*, *Freeman4*,*Coke*, and *Woman*). In each row, a different baseline tracker is applied (from top to bottom: SAMF, KCF, DCF, CSK, and MOSSE) along with its adaptive target variant.

References

1. Avidan, S.: Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2007.
2. Avidan, S.: Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2004.
3. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009.
4. Bibi, A., Ghanem, B.: Multi-template scale-adaptive kernelized correlation filters. In: IEEE International Conference on Computer Vision Workshops, ICCVW, 2015.
5. Bolme, D.S., Beveridge, J.R., Draper, B., Lui, Y.M., et al.: Visual object tracking using adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010.
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2003
7. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, BMVC, 2014.
8. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: IEEE International Conference on Computer Vision, ICCV, 2015.
9. Danelljan, M., Khan, F., Felsberg, M., Weijer, J.: Adaptive color attributes for real-time visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014.
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision, ECCV, 2012.
11. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2015.
12. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.
13. Kiani Galoogahi, H., Sim, T., Lucey, S.: Correlation filters with limited boundaries. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.
14. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: European Conference on Computer Vision Workshops, ECCV, 2014.
15. Liu, T., Wang, G., Yang, Q.: Real-time part-based visual tracking via adaptive correlation filters. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.
16. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: IEEE International Conference on Computer Vision, ICCV, 2009.
17. Poggio, T., Cauwenberghs, G.: Incremental and decremental support vector machine learning. In Advances in Neural Information Processing Systems, NIPS, 2001
18. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2015.
19. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2013.

20. Zhang, J., Ma, S., Sclaroff, S.: Meem: Robust tracking via multiple experts using entropy minimization. In: European Conference on Computer Vision, ECCV, 2014.
21. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Low-rank sparse learning for robust visual tracking. In: European Conference on Computer Vision, ECCV, 2012.
22. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2012.
23. Zhang, T., Liu, S., Xu, C., Yan, S., Ghanem, B., Ahuja, N., Yang, M.H.: Structural sparse tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.