Context-Aware Correlation Filter Tracking

Matthias Mueller, Neil Smith, Bernard Ghanem
King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
{matthias.mueller.2, neil.smith, bernard.ghanem}@kaust.edu.sa

Abstract

Correlation filter (CF) based trackers have recently gained a lot of popularity due to their impressive performance on benchmark datasets, while maintaining high frame rates. A significant amount of recent research focuses on the incorporation of stronger features for a richer representation of the tracking target. However, this only helps to discriminate the target from background within a small neighborhood. In this paper, we present a framework that allows the explicit incorporation of global context within CF trackers. We reformulate the original optimization problem and provide a closed form solution for single and multi-dimensional features in the primal and dual domain. Extensive experiments demonstrate that this framework significantly improves the performance of many CF trackers with only a modest impact on frame rate.

1. Introduction

Object tracking remains a core problem in computer vision with numerous applications, such as surveillance, human-machine interaction, robotics, etc. Large new datasets and benchmarks such as OTB-50 [26], OTB-100 [27], TC-128 [19], ALOV300++ [24] and UAV123 [23], as well as, tracking challenges such as the visual object tracking (VOT) challenge and multi-object tracking (MOT) challenge have sparked the interest of many researchers and helped advance the field significantly. Despite substantial progress in recent years, visual object tracking remains a challenging problem in computer vision.

In this paper, we address the problem of single-object tracking, which is commonly approached as a tracking-by-detection problem. Currently, most research focuses on model-free generic object trackers, where no prior assumptions regarding the object appearance are made. The generic nature of this problem makes it challenging, since there are very few constraints on object appearance, and the object can undergo a variety of unpredictable transformations in consecutive frames (*e.g.* aspect ratio change, illumination variation, in/out-of-plane rotation, occlusion, etc.).

The tracking problem can be divided into two main challenges, object representation and sampling for detection. Recently, most successful single-object tracking algorithms use a discriminative object representation with either strong hand-crafted features, such as HOG and Colornames, or learned ones. Recent work has integrated deep features [21] trained on a large dataset, such as ImageNet, to represent the tracked object. Sampling on the other hand is a trade-off between computation time and precise scanning of the region of interest for the target.

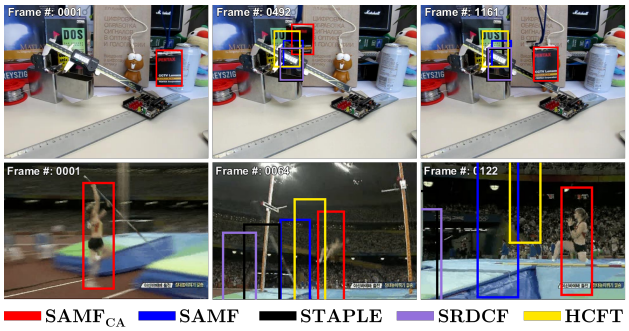


Figure 1: Tracking results of our context-aware adaptation of the baseline SAMF tracker, denoted as SAMF$_{CA}$, and a comparison with recent state-of-the-art tracking algorithms on the *Box* and *Jump* sequences from OTB-100.

Lately, CF trackers have sparked a lot of interest, due to their high accuracy while running at high frame rates. [4, 6, 10, 11, 14]. In general, CF trackers learn a correlation filter online to localize the object in consecutive frames. The learned filter is applied to the region of interest in the next frame and the location of the maximum response corresponds to the object location. The filter is then updated by using the new object location. The major reasons behind the success of this tracking paradigm is the approximate dense sampling performed by circularly shifting the training samples and the computational efficiency of learning the correlation filter in the Fourier domain. Provided that the background is homogeneous and the object does not move much, these circular shifts are equivalent to actual translations in the image and this framework works well.

However, since these assumptions are not always valid, CF trackers have several drawbacks. One major drawback is

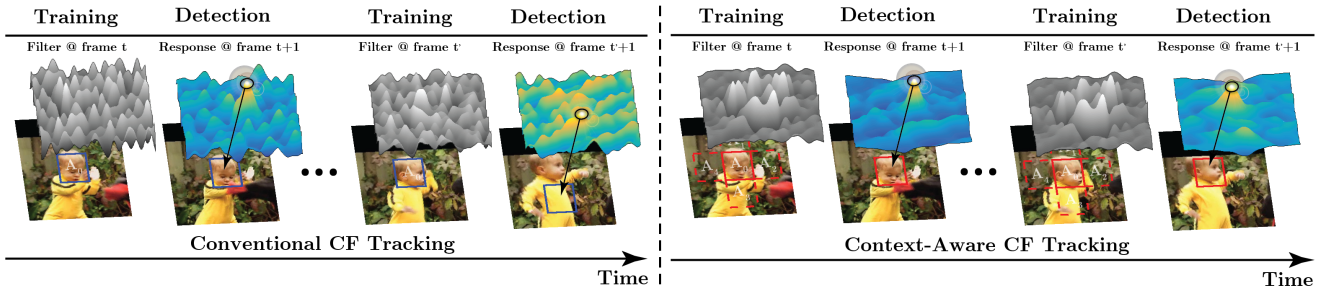


Figure 2: Comparing conventional CF tracking to our proposed context-aware CF tracking.

that there are boundary effects due to the circulant assumption. In addition, the target search region only contains a small local neighborhood to limit drift and keep computational cost low. The boundary effects are usually suppressed by a cosine window, which effectively reduces the search region even further. Therefore, CF trackers usually have very limited information about their context and easily drift in cases of fast motion, occlusion or background clutter. In order to address this limitation, we propose a framework that takes global context into account and incorporates it directly into the learned filter (see Figure 2). We derive a closed-form solution for our formulation and propose it as a framework that can be easily integrated with most CF trackers to boost their performance, while maintaining their high frame rate. As shown in Figure 1, integrating our framework with the mediocre tracker SAMF [18] achieves better tracking results than state-of-the-art trackers by exploiting context information. Note, that it even outperforms the very recent HCFT tracker [21], whose hierarchical convolutional features implicitly contain context information. We show through extensive evaluation on several large datasets that integrating our framework improves all tested CF trackers and allows top-performing CF trackers to exceed current state-of-the-art precision and success scores on the well-known OTB-100 benchmark [27].

2. Related Work

CF Trackers. Since the MOSSE work of Bolme *et al*. [4], correlation filters (CF) have been studied as a robust and efficient approach to the problem of visual tracking. Major improvements to MOSSE include the incorporation of kernels and HOG features [10], the addition of color name features [18] or color histograms [1], integration with sparse tracking [30], adaptive scale [2, 5, 18], mitigation of boundary effects [6], and the integration of deep CNN features [21]. Currently, CF-based trackers rank at the top of current benchmarks, such as OTB-100 [27], UAV123 [23], and VOT2015 [17], while remaining computationally efficient.

CF Variations and Improvements. Significant attention in recent work has focused on extending CF trackers to address inherent limitations. For instance, Liu *et al*. propose part-based tracking to reduce sensitivity to partial occlusion and better preserve object structure [20]. The work of [22] performs long term-tracking that is robust to appearance variation by correlating temporal context and training an online random fern classifier for re-detection. Zhu *et al*. propose a collaborative CF that combines a multi-scale kernelized CF to handle scale variation with an online CUR filter to address target drift [31]. These approaches register improvements by either combining external classifiers to assist the CF or taking advantage of its high computational speed to run multiple CF trackers at once.

CF Frameworks. Recent work [2, 3] has found that some of these inherent limitations can be overcome directly by modifying the conventional CF model used for training. For example, by adapting the target response (used for ridge regression in CF) as part of a new formulation, Bibi *et al*. significantly decrease target drift while remaining computationally efficient [3]. This method yields a closed-form solution and can be applied to many CF trackers as a framework. Similarly, this paper also proposes a framework that makes CF trackers context-aware and increases their performance beyond the improvement attainable by [3], while being less computationally expensive.

Context Trackers. The use of context for tracking has been explored in previous work by Dinah *et al*. [7], where distractors and supporters are detected and tracked using a sequential randomized forest, an online template-based appearance model, and local features. In more recent work, contextual information of a scene is exploited using a multi-level clustering to detect similar objects and other potential distractors [28]. A global dynamic constraint is then learned online to discriminate these distractors from the object of interest. This approach shows improvement on a subset of cluttered scenes in OTB-100, where distractors are predominant. However, both of these trackers do not generalize well and, as a result, their overall performance on current benchmarks is only average. In contrast, our approach is more generic and can make use of varying types of contextual image regions that may or may not contain distractors. We show that context awareness enables improvement across the entire OTB-100 and is not limited to cluttered scenes, where context can lead to the most improvement.

Contributions. To the best of our knowledge, **(i)** this is the first context-aware formulation that can be applied as a framework to most CF trackers. Its closed form solution al-

lows CF trackers to remain computationally efficient, while significantly improving their performance. **(ii)** Extensive experiments on several datasets show the effectiveness of our formulation. All CF trackers benefit from a boost in performance, while remaining computationally efficient.

3. CF Tracking

Before the detailed discussion of our proposed framework and for completeness, we first revisit the details of conventional CF tracking. CF trackers use discriminative learning at their core. The goal is to learn a discriminative correlation filter that can be applied to the region of interest in consecutive frames to infer the location of the target (*i.e*. location of maximum filter response). The key contribution leading to the popularity and success of CF trackers is their sampling method. Due to computational constraints, it is common practice to randomly pick a limited number of negative samples around the target. The sophistication of the sampling strategy and the number of negative samples can have a significant impact on tracking performance. CF trackers allow for dense sampling around the target at very low computational cost. This is achieved by modeling all possible translations of the target within a search window as circulant shifts and concatenating them to form the data matrix $\mathbf{A}_0$. The circulant structure of this matrix facilitates a very efficient solution to the following ridge regression problem in the Fourier domain.

$$\min_{\mathbf{w}} \ ||\mathbf{A}_0\mathbf{w} - \mathbf{y}||_2^2 + \lambda_1||\mathbf{w}||_2^2 \tag{1}$$

Here, the learned correlation filter is denoted by the vector $\mathbf{w}$. The square matrix $\mathbf{A}_0$ contains all circulant shifts of the vectorized image patch $\mathbf{a}_0$ and the regression target $\mathbf{y}$ is a vectorized image of a 2D Gaussian.

Notation. We denote the j^{th} component of vector $\mathbf{x}$ as $\mathbf{x}(j)$. We denote its conjugate by $\mathbf{x}^*$ and its Fourier transform $\mathbf{F}^H\mathbf{x}$ by $\hat{\mathbf{x}}$, where $\mathbf{F}$ is the DFT matrix. The following identity for circulant matrices is the key ingredient for solving Eq. (1) efficiently:

$$\mathbf{X} = \mathbf{F}\ \mathrm{diag}(\hat{\mathbf{x}})\ \mathbf{F}^H \ \text{ and } \ \mathbf{X}^T = \mathbf{F}\ \mathrm{diag}(\hat{\mathbf{x}}^*)\ \mathbf{F}^H \tag{2}$$

3.1. Solution in the Primal Domain

The objective in Eq. (1) is convex and has a unique global minimum. Equating its gradient to zero leads to a closed-form solution for the filter: $\mathbf{w} = (\mathbf{A}_0^T\mathbf{A}_0 + \lambda_1\mathbf{I})^{-1}\mathbf{A}_0^T\mathbf{y}$. Since $\mathbf{A}_0$ is circulant, it can be diagonalized using Eq. (2) and matrix inversion can be done efficiently in the Fourier domain [10]:

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1} \tag{3}$$

Detection formula. The learned filter $\mathbf{w}$ is convolved with image patch $\mathbf{z}$ (search window) in the next frame, where $\mathbf{Z}$ denotes its circulant matrix. The location of the maximum response is the target location within the search window. The primal detection formula is given by:

$$\mathbf{r}_p(\mathbf{w}, \mathbf{Z}) = \mathbf{Z}\ \mathbf{w} \ \Leftrightarrow \ \hat{\mathbf{r}}_p = \hat{\mathbf{z}} \odot \hat{\mathbf{w}} \tag{4}$$

3.2. Solution in the Dual Domain

Eq. (1) can also be solved in the dual domain using the dual variable $\boldsymbol{\alpha}$, which relates to the primal variable through $\mathbf{w} = \mathbf{A}_0^T\boldsymbol{\alpha}$. The dual closed-form solution is: $\boldsymbol{\alpha} = (\mathbf{A}_0\mathbf{A}_0^T)^{-1}\mathbf{y}$. Similar to the primal domain, it can be computed efficiently in the Fourier domain [10]:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1} \tag{5}$$

Since the solution can be written as a function of bi-products, the kernel trick can also be applied allowing the use of kernels in the dual domain [11].

Detection formula. The dual variable $\boldsymbol{\alpha}$ can be used directly for detection by expressing it in terms of the primal variable. This leads to the following dual detection formula:

$$\mathbf{r}_d(\boldsymbol{\alpha}, \mathbf{A}_0, \mathbf{Z}) = \mathbf{Z}\ \mathbf{A}_0^T\boldsymbol{\alpha} \ \Leftrightarrow \ \hat{\mathbf{r}}_d = \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_0^* \odot \hat{\boldsymbol{\alpha}} \tag{6}$$

4. Context-Aware CF Tracking

The surroundings of the tracked object can have a big impact on tracking performance. For example, if there is a lot of background clutter, context is very important for successful tracking. Therefore, we propose a framework for CF trackers that adds contextual information to the filter during the learning stage (Figure 2).

In every frame, we sample k context patches $\mathbf{a}_i \in \mathbb{R}^n$ around the object of interest $\mathbf{a}_0 \in \mathbb{R}^n$ according to the sampling strategy in Sec. 4.3. Their corresponding circulant matrices are $\mathbf{A}_i \in \mathbb{R}^{n\times n}$ and $\mathbf{A}_0 \in \mathbb{R}^{n\times n}$, respectively. These context patches can be viewed as hard negative samples. They contain global context in the form of various distractors and diverse background. Intuitively speaking, we want to learn a filter $\mathbf{w} \in \mathbb{R}^n$ that has a high response for the target patch and close to zero response for context patches (Figure 2). We encourage this by adding the context patches as a regularizer to the standard formulation (see Eq. (7)). As a result, the target patch is regressed to $\mathbf{y}$ like in the standard formulation (Eq. (1)), while the context patches are regressed to zeros controlled by the parameter λ_2.

$$\min_{\mathbf{w}} \ \|\mathbf{A}_0\mathbf{w} - \mathbf{y}\|_2^2 + \lambda_1\|\mathbf{w}\|_2^2 + \lambda_2\sum_{i=1}^{k}\|\mathbf{A}_i\mathbf{w}\|_2^2 \tag{7}$$

Note, that there are other possible choices for incorporating the context term (*e.g*. hinge loss). This would enforce a

lower response at context patches than the target, rather than regressing them to zero, which is perhaps a better assumption. However, it leads to a constrained convex optimization requiring an iterative solution, which is quite slow.

4.1. Single-Channel Features

Solution in the Primal Domain. The primal objective function f_p in Eq. (7) can be rewritten by stacking the context image patches below the target image patch forming a new data matrix $\mathbf{B} \in \mathbb{R}^{(k+1)n \times n}$. The new regression target $\bar{\mathbf{y}} \in \mathbb{R}^{(k+1)n}$ concatenates $\mathbf{y}$ with zeros.

$$f_p(\mathbf{w}, \mathbf{B}) = \|\mathbf{B}\mathbf{w} - \bar{\mathbf{y}}\|_2^2 + \lambda_1 ||\mathbf{w}||_2^2 \tag{8}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_0 \\ \sqrt{\lambda_2}\mathbf{A}_1 \\ \vdots \\ \sqrt{\lambda_2}\mathbf{A}_k \end{bmatrix} \text{ and } \bar{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

Since $f_p(\mathbf{w}, \mathbf{B})$ is convex, it can be minimized by setting the gradient to zero, yielding:

$$\mathbf{w} = \left(\mathbf{B}^T\mathbf{B} + \lambda_1 \mathbf{I}\right)^{-1} \mathbf{B}^T \bar{\mathbf{y}} \tag{9}$$

Similar to the CF tracker in Eq. (1), we use Eq. (2) to obtain the following closed-form solution in the Fourier domain.

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^{k} \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{a}}_i} \tag{10}$$

Detection formula. It is exactly the same as in the standard formulation in Eq. (4).

Solution in the Dual Domain. Note that the solution in the primal domain in Eq. (9) has the exact same form as the solution of the standard ridge regression problem [11]. Hence, the solution in the dual domain is given by:

$$\boldsymbol{\alpha} = \left(\mathbf{B}\mathbf{B}^T + \lambda_1 \mathbf{I}\right)^{-1} \bar{\mathbf{y}}, \text{ where } \boldsymbol{\alpha} \in \mathbb{R}^{(k+1)n} \tag{11}$$

Using the identity for circulant matrices yields:

$$\hat{\boldsymbol{\alpha}} = \begin{bmatrix} \operatorname{diag}(\mathbf{d}_{00}) & \dots & \operatorname{diag}(\mathbf{d}_{0k}) \\ \vdots & \ddots & \vdots \\ \operatorname{diag}(\mathbf{d}_{k0}) & \dots & \operatorname{diag}(\mathbf{d}_{kk}) \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{y}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{12}$$

where vectors $\mathbf{d}_{jl}$ with $j, l \in \{1, ..., k\}$ are given by:

$$\begin{cases} \mathbf{d}_{00} = \hat{\mathbf{a}}_0 \odot \hat{\mathbf{a}}_0^* + \lambda_1 \\ \mathbf{d}_{jj} = \lambda_2 \left(\hat{\mathbf{a}}_j \odot \hat{\mathbf{a}}_j^*\right) + \lambda_1, \; j \neq 0 \\ \mathbf{d}_{jl} = \sqrt{\lambda_2} \left(\hat{\mathbf{a}}_j \odot \hat{\mathbf{a}}_l^*\right), \; j \neq l \end{cases} \tag{13}$$

Note that the kernel trick can be applied, since all interactions between the image patches occur as bi-products. Hence, the linear correlation can simply be replaced by one of the kernel correlations as derived for conventional kernelized CF trackers [10].

Since all blocks are diagonal, the system can be decomposed into n smaller systems of dimension $\mathbb{R}^{(k+1)\times(k+1)}$. This significantly reduces complexity and allows for parallelization. Instead of solving one large system of dimension $\mathbb{R}^{(k+1)n\times(k+1)n}$ to compute $\hat{\boldsymbol{\alpha}}$, a separate system is solved for each pixel $p \in \{1, ..., n\}$ of $\hat{\boldsymbol{\alpha}}$, as follows:

$$\hat{\boldsymbol{\alpha}}(p) = \begin{bmatrix} \mathbf{d}_{00}(p) & \dots & \mathbf{d}_{0k}(p) \\ \vdots & \ddots & \vdots \\ \mathbf{d}_{k0}(p) & \dots & \mathbf{d}_{kk}(p) \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{y}}(p) \\ \vdots \\ 0 \end{bmatrix} \tag{14}$$

Detection formula. Note that the detection formula for the dual domain in Eq. (6) needs to be adapted to our formulation: $\mathbf{r}_d(\boldsymbol{\alpha}, \mathbf{B}, \mathbf{Z}) = \mathbf{Z}\,\mathbf{B}^T\boldsymbol{\alpha}$. It is similar to the standard formulation, but $\mathbf{B}$ contains the context patches in addition to the target. Consequently, $\boldsymbol{\alpha} \in \mathbb{R}^{(k+1)n}$ is now composed of a concatenation of dual variables $\{\boldsymbol{\alpha}_0, \dots, \boldsymbol{\alpha}_k\}$. After diagonalization using Eq. (2), the detection formula can be rewritten as follows in the Fourier domain:

$$\hat{\mathbf{r}}_d = \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_0^* \odot \hat{\boldsymbol{\alpha}}_0 + \sqrt{\lambda_2} \sum_{i=1}^{k} \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_i^* \odot \hat{\boldsymbol{\alpha}}_i \tag{15}$$

4.2. Multi-Channel Features

Solution in the Primal Domain. Since multi-channel features usually offer a much richer representation of the target than single-channel features (*e.g.* grayscale intensity), it is important to generalize Eq. (7) to multi-channel features and learn a joint filter for all feature dimensions m. The multi-channel primal objective function $f_p(\bar{\mathbf{w}}; \bar{\mathbf{B}})$ can be written in a similar fashion as in the case of single-channel features (Eq. (8)), but with the following differences: $\bar{\mathbf{B}} \in \mathbb{R}^{(k+1)n \times nm}$ now contains the target and context image patches as rows and their corresponding features as columns. The filters for the different feature dimensions are stacked into $\bar{\mathbf{w}} \in \mathbb{R}^{nm}$.

$$f_p(\bar{\mathbf{w}}) = \left\|\bar{\mathbf{B}}\bar{\mathbf{w}} - \bar{\mathbf{y}}\right\|_2^2 + \lambda_1 ||\bar{\mathbf{w}}||_2^2 \tag{16}$$

Minimizing Eq. (16) is similar to the single-channel case:

$$\bar{\mathbf{w}} = (\bar{\mathbf{B}}^T\bar{\mathbf{B}} + \lambda_1 \mathbf{I})^{-1} \bar{\mathbf{B}}^T \bar{\mathbf{y}} \tag{17}$$

Using the identity for circulant matrices yields:

$$\hat{\bar{\mathbf{w}}} = \begin{bmatrix} \bar{\mathbf{C}}_{11} & \dots & \bar{\mathbf{C}}_{1m} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{C}}_{m1} & \dots & \bar{\mathbf{C}}_{mm} \end{bmatrix}^{-1} \begin{bmatrix} \operatorname{diag}(\hat{\mathbf{a}}_{01}^* \odot \hat{\mathbf{y}}) \\ \vdots \\ \operatorname{diag}(\hat{\mathbf{a}}_{0m}^* \odot \hat{\mathbf{y}}) \end{bmatrix}$$

The target and context image patches for each feature dimension $j, l \in \{1, ..., m\}$ are denoted by $\mathbf{a}_{0j}$ and $\mathbf{a}_{ij}$ respectively. The blocks of $(\bar{\mathbf{B}}^T\bar{\mathbf{B}} + \lambda_1\mathbf{I})^{-1}$ are defined as:

$$\begin{cases} \bar{\mathbf{C}}_{jj} & = \text{diag}\left(\hat{\mathbf{a}}_{0j}^* \odot \hat{\mathbf{a}}_{0j} + \lambda_2 \sum_{i=1}^{k} \hat{\mathbf{a}}_{ij}^* \odot \hat{\mathbf{a}}_{ij}\right) + \lambda_1\mathbf{I} \\ \bar{\mathbf{C}}_{jl} & = \text{diag}\left(\hat{\mathbf{a}}_{0j}^* \odot \hat{\mathbf{a}}_{0l} + \lambda_2 \sum_{i=1}^{k} \hat{\mathbf{a}}_{ij}^* \odot \hat{\mathbf{a}}_{il}\right),\ j \neq l \end{cases} \tag{18}$$

Unfortunately, this system cannot be inverted as efficiently as in the single-channel case (Eq. (10)). However, since all of the blocks are diagonal, the system can be decomposed into n smaller systems of dimension $\mathbb{R}^{m\times m}$. This reduces the complexity significantly and allows for parallelization. Similar to Eq. (14), a separate system is solved for each pixel $p \in \{1, ..., n\}$ of the filter $\hat{\bar{\mathbf{w}}}$.

Detection formula. It is almost the same as in the standard formulation in Eq. (4) with the difference that the image patch $\mathbf{z}$ and the learned filter $\mathbf{w}$ are m-dimensional.

Solution in the Dual Domain. Just like in the case of single-channel features, the multi-channel primal solution (Eq. (17)) also has the exact same form as the solution of the standard ridge regression problem yielding the following solution in the dual domain:

$$\bar{\boldsymbol{\alpha}} = \left(\bar{\mathbf{B}}\bar{\mathbf{B}}^T + \lambda_1\mathbf{I}\right)^{-1} \bar{\mathbf{y}} \ \text{ where } \bar{\boldsymbol{\alpha}} \in \mathbb{R}^{kn} \tag{19}$$

Again, the identity for circulant matrices (Eq. (2)) yields:

$$\hat{\bar{\boldsymbol{\alpha}}} = \begin{bmatrix} \text{diag}(\bar{\mathbf{d}}_{00}) & \dots & \text{diag}(\bar{\mathbf{d}}_{0k}) \\ \vdots & \ddots & \vdots \\ \text{diag}(\bar{\mathbf{d}}_{k0}) & \dots & \text{diag}(\bar{\mathbf{d}}_{kk}) \end{bmatrix}^{-1} \begin{bmatrix} \hat{\mathbf{y}} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{20}$$

where vectors $\bar{\mathbf{d}}_{jl}$ with $j, l \in \{1, ..., k\}$ are given by:

$$\begin{cases} \bar{\mathbf{d}}_{00} & = \sum_{i=1}^{m} (\hat{\mathbf{a}}_{0i} \odot \hat{\mathbf{a}}_{0i}^*) + \lambda_1 \\ \bar{\mathbf{d}}_{jj} & = \lambda_2 \sum_{i=1}^{m} (\hat{\mathbf{a}}_{ji} \odot \hat{\mathbf{a}}_{ji}^*) + \lambda_1,\ j \neq 0 \\ \bar{\mathbf{d}}_{jl} & = \sqrt{\lambda_2} \sum_{i=1}^{m} (\hat{\mathbf{a}}_{ji} \odot \hat{\mathbf{a}}_{li}^*),\ j \neq l \end{cases} \tag{21}$$

Note that the linear system is the same as in case of the dual domain solution for single-channel features (Sec. 4.1) with the exception that there is now a sum along the feature dimension m. This solution also permits the use of kernels and the linear system can be solved in the same fashion as the single-channel case (Eq. (14)).

Detection formula. It follows the single-channel feature case with the difference that $\bar{\mathbf{Z}} \in \mathbb{R}^{nm\times n}$ and $\bar{\mathbf{B}} \in \mathbb{R}^{(k+1)n\times nm}$ now have multiple feature dimensions as columns: $\mathbf{r}_d(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{B}}, \bar{\boldsymbol{Z}}) = \bar{\mathbf{Z}}\,\bar{\mathbf{B}}^T\bar{\boldsymbol{\alpha}}$. After diagonalization and rearrangement of terms, the detection formula in the Fourier domain reduces to its final form:

$$\hat{\mathbf{r}}_d = \sum_{i=1}^{m} \hat{\mathbf{z}}_i \odot \hat{\mathbf{a}}_{0i}^* \odot \hat{\bar{\boldsymbol{\alpha}}}_0 + \sqrt{\lambda_2} \sum_{j=1}^{k} \sum_{i=1}^{m} \hat{\mathbf{z}}_i \odot \hat{\mathbf{a}}_{ji}^* \odot \hat{\bar{\boldsymbol{\alpha}}}_j \tag{22}$$

4.3. Implementation Details

Generalization. We derived a closed-form solution for all combinations of single-channel/multi-channel features and primal/dual domain. Consequently, our framework can be broadly applied to many different types of CF trackers, thus, impacting performance across the board. In the case of single-channel features in the primal domain, the solution only contains element-wise operations and the implementation is trivial. However, in the remaining cases, a linear system needs to be solved efficiently. For single-channel features in the dual domain, we need to invert n small systems of size $(k+1)\times(k+1)$. Each of them can be re-written as an outer product and can be solved very efficiently using the Sherman-Morrison formula for inversion [9, 16].

Since the solution of the multi-channel problem contains a sum over patches in both primal and dual domain, it cannot be rewritten as an outer product. Instead, either one large system or n smaller systems have to be solved. To solve the large system, conjugate gradient descent (CGD) can be used. The smaller systems can be solved exactly. The choice depends on the number of feature dimensions m in case of the primal domain or the number of context patches k in case of the dual domain. The complexity of solving n smaller systems is always lower and the systems are dense. If m or k is sufficiently small, it is more efficient to solve those systems directly.

However, since the large system (primal: $nm \times nm$ and dual: $(k+1)n \times (k+1)n$) is sparse, one can use CGD, if m or k is very large. In the case of multi-channel features in the primal domain and assuming the m features are independent (*e.g.* when HOG is used, the filter for each feature dimension can be computed independently using Eq. (9). A similar assumption for the target patch and k context patches cannot be made in the dual domain. If they are independent, the problem reduces to the regular ridge regression problem in Eq. (1) for multi-channel features.

Context Selection Strategy. The selection of context patches is essential for tracking performance.There are several strategies to do this selection. The most naive and generic approach is to simply sample context patches uniformly around the target. Essentially, this will equip the filter to better discriminate against context that will probably become background or an occluder in the near future. Moreover, a Kalmann filter with a constant velocity model can be used to guide the sampling. Another approach that can complement previous strategies or be used on its own is in the spirit of hard negative mining [15]. At each frame, context patches are sampled at locations, where the filter response is high and spatially far from the maximum. In this case, the filter learns to have a low response at image patches that look similar to the target. Likewise, the sampling strategy could be adapted for multi-object tracking (*i.e.* initialize a single object tracker for each object),

where context patches are taken to be the object detections at each frame that are not overlapping with the target.

4.4. Comparison with CF Trackers

Conventional CF tracking [4, 10, 11]**.** A naive approach for enlarging the scope of these CF trackers is to use a bigger search window. If at all, this only would help during the detection stage, as the training stage remains the same. However, in practice, a larger search window introduces more possible distractors increasing the chance of drift, if the trained filter is not be representative enough. In contrast, our framework takes global context into account during the training stage. Hence, the learned filter is more discriminative in the current context and as a result more robust to drift due to fast motion, occlusion and distractors. Moreover, even if the feature representation is not rich, the tracker might still be able to track the object by focusing on context information, *i.e*. the filter response indicates where the target should not be.

Target-adaptive CF tracking. Target-adaptive CF tracking [3] is also a framework that can be applied to CF trackers to improve their performance. However, this work focuses on carefully designing a better $\mathbf{y}$ in order to address boundary effects and deal with fast motion and occlusion scenarios. While our work also improves tracking performance in these scenarios, context is also helpful in many other situations (*e.g*. when drastic appearance change, background clutter, and illumination variation occur). The results in Sec. 5.3 show that trackers using our framework achieve better overall performance than using the target-adaptive one, while running at much higher frame-rates. Ideally, both frameworks can be combined with a potential of improving performance; however, we aim to keep runtime within acceptable bounds. As such, combining them falls outside the scope of this paper.

5. Experiments

To validate the effectiveness of our framework, we integrate it with four popular and diverse CF trackers. We then benchmark them against their baseline versions and the target-adaptive framework in [3]. For evaluation, we use the popular object tracking benchmark OTB-100 [27].

5.1. Baseline Trackers

In order to represent the realm of CF tracking, we select a wide variety of recent CF trackers as baselines. We only select trackers that follow the standard CF formulation (Eq. (1)) and are not too similar to other selected baselines in terms of features and/or implementation. Table 1 summarizes these CF trackers. We apply our framework to these selected baselines and call them MOSSE_{CA}, DCF_{CA}, SAMF_{CA}, and $\text{STAPLE}_{\text{CA}}$. Also, we include their target-adaptive counterparts (if available) and refer to them as MOSSE_{AT}, DCF_{AT}, and SAMF_{AT}.

Table 1: Baseline CF trackers to be added to our framework

Tracker	Features	Scale	Published
MOSSE	grayscale pixel intensity	No	2010 (CVPR)
DCF	HOG [8]	No	2015 (TPAMI)
SAMF	HOG [8], CN [25]	Yes	2014 (ECCV-W)
STAPLE	HOG [8], color histogram	Yes	2016 (CVPR)

5.2. Experiment Setup

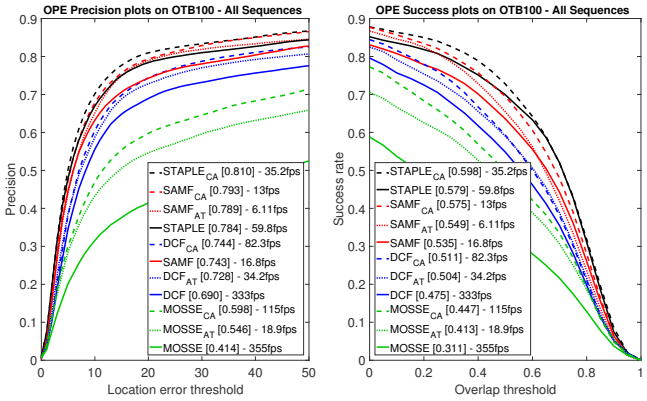
Evaluation Methodology. All trackers are evaluated according to two measures, precision and success, as defined in OTB-50/OTB-100 [26, 27]. Precision measures the center error between tracker bounding box and ground truth bounding box. In the precision plot, the maximum allowed center error in pixel distance is varied along the x-axis and the percentage of correctly predicted bounding boxes per threshold is plotted on the y-axis. The common threshold of 20 pixels [26] is used for ranking trackers. Success is measured as the intersection over union (IoU) of tracker bounding box and groundtruth bounding box. In the success plot, the required overlap is varied along the x-axis and the percentage of correctly predicted bounding boxes per threshold is plotted on the y-axis. Trackers are ranked by the area under the curve (AUC) [26]. We include the precision plot on OTB-100 for reference, but focus on success plots for conclusions and more detailed analysis, since it is more indicative of actual tracking performance. All trackers are run on the same workstation (Intel Xeon CPU E5-2697 2.6GHz, 256GB RAM) using MATLAB.

Parameter Settings. All baseline trackers and adaptive target trackers are run with the standard parameters provided by the authors. For fair comparison, we run the context-aware trackers with the same parameters. We set the additional regularization factor λ_2 to $\{2, 25, 0.4, 0.5\}$ and use the standard update rule with learning rate $\{0.025, 0.015, 0.005, 0.015\}$ for MOSSE_{CA}, DCF_{CA}, SAMF_{CA} and $\text{STAPLE}_{\text{CA}}$, respectively. We set the number of context patches k to 4 and sample them uniformly around the target. Increasing k beyond 4, only results in minor improvement but affects runtime. We also increase the padding for all *CA* trackers due to the increased robustness from context.

5.3. Quantitative Results

Figure 3a and 3b show the results of all baseline trackers and their adaptive target and context-aware counterparts on OTB-100. All *CA* trackers improve their baseline. The performance gain decreases as more sophisticated features are used. The absolute improvement for precision and success ranges from $\{2.6\%, 2.1\%\}$ to $\{18.4\%, 13.6\%\}$ for the most

sophisticated CF tracker (STAPLE) and the most basic CF tracker (MOSSE) respectively. In addition, our proposed framework does not only outperform the baseline, but also the corresponding *AT* trackers (not available for STAPLE). Note that this performance gain is achieved at a much lower computation cost compared to the adaptive target framework. The *CA* trackers run at approximately half the speed of the baseline trackers but 2-6 times faster than their *AT* counterparts.

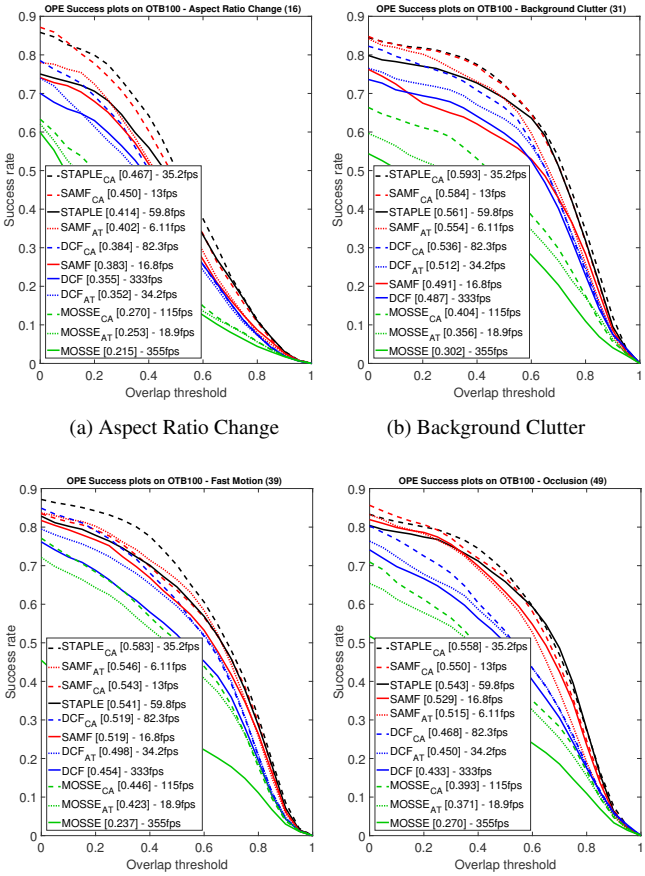


(a) OTB-100 - Precision Plot (b) OTB-100 - Success Plot

Figure 3: Average overall performance on OTB-100

Evaluation per attribute. While our framework improves tracking performance in most scenarios there are certain categories that benefit more than others. The most significant improvement is achieved in the cases of drastic aspect ratio change (Figure 4a), background clutter (Figure 4b), fast motion (Figure 4c) and occlusion (Figure 4d). In particular, if the object appearance changes drastically (*e.g.* aspect ratio change, occlusion) or if the background looks similar to the target (*e.g.* background clutter), our framework is very beneficial. Furthermore, our framework also improves significantly for videos with fast motion. This is largely due to the fact that adding the context discrimination allows for a larger search region. It is also notable that in most cases our method outperforms the adaptive target framework, which is designed specifically to improve performance for fast motion scenarios. There are several more categories, where our method excels including motion blur, deformation, illumination variation, in/out-of-plane rotation, etc. Please refer to the **supplementary material** for all per-attribute results.

Comparison to state-of-the-art trackers. To put the tracking performance into perspective, we compare the best performing context-aware CF trackers (SAMF_{CA} and $\text{STAPLE}_{\text{CA}}$) and their baselines (SAMF [18] and STAPLE [1]) to the most recent state-of-the-art trackers (SOWP [15], HCFT [21], and MEEM [29]), which are not necessarily CF



(a) Aspect Ratio Change (b) Background Clutter

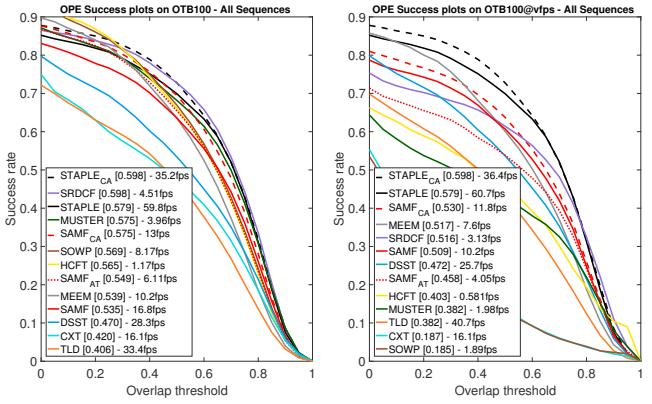
(c) Fast Motion (d) Occlusion

Figure 4: Average performance on OTB-100 for 4 attributes

based. In addition, we include popular CF trackers that did not meet the selection criteria for our framework, namely DSST [5] that is very similar to SAMF but usually underperforms it, MUSTER [12] that employs a long-term/short-term memory strategy but its source code is not available for modification, and SRDCF [6] that ranks first on most recent tracking benchmarks but does not follow the standard formulation. Lastly, we include TLD [13] and the classic context tracker CXT [7] for reference. As Figure 5a shows, our framework enables $\text{STAPLE}_{\text{CA}}$ to emerge as the top performer among the latest state-of-the-art trackers.

Variable frame rate evaluation. In order to demonstrate the computational efficiency of our framework, we downsample the OTB100 dataset for each tracker according to their actual speed. The results in Figure 5b show that most state-of-the-art trackers are very slow and as a result their performance degrades significantly, when the frame rate is adapted according to actual tracking speed. In contrast, the performance of our context-aware CF trackers only degrades marginally solidifying their rank as top performers. Comparing the target-adaptive framework to our context-

aware framework in the case of SAMF is another indicator for its efficiency. While SAMF_{AT} drops below its baseline, the context-aware counterpart SAMF_{CA} is able to maintain its edge. We believe that this comparison is important to gauge the way trackers will perform in real-world online tracking scenarios. As such, we encourage the community to make such a comparison a standard evaluation criterion.



(a) All Frames (b) Variable Frame Rate

Figure 5: Average performance of state-of-the-art trackers on OTB-100 with and without variable frame rate

5.4. Qualitative Results

To visualize the impact our framework has on tracking performance, we show examples of each baseline method compared to its context-aware counterpart on sample videos from OTB-100 in Figure 6.

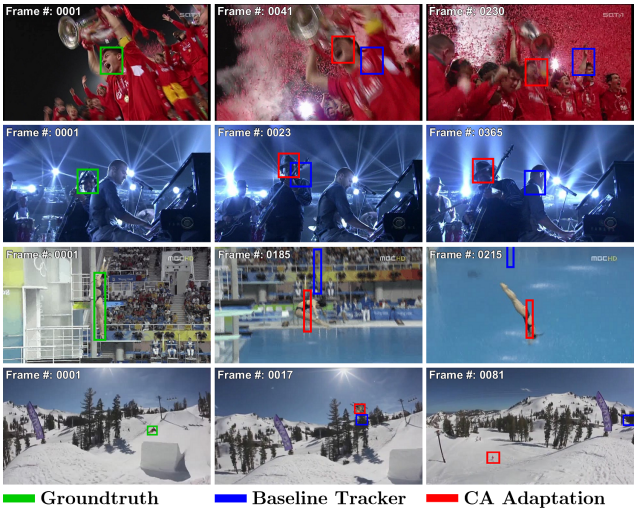


Figure 6: Tracking results of four baseline CF trackers compared to their context-aware counterparts. The trackers and corresponding videos are (from top to bottom): MOSSE: *Soccer*, DCF: *Shaking*, SAMF: *Diving*, STAPLE: *Skiing*.

5.5. Discussion

There are few situations, where our framework provides little benefit, but this does occur when targets are low resolution. In this case, our framework still improves the baseline tracker, but it tends to be outperformed by the adaptive target framework. Being aware of the context by incorporating more negative samples helps, but it is not the most effective approach in this scenario. Since the target representation is weak (low resolution), it is more helpful to incorporate additional positive samples, which is implicitly done in the adaptive target framework, when samples are added for learning the best regression target $\mathbf{y}$ [3].

Depending on the selection of single-channel/multi-channel features and primal/dual domain solution, the complexity of the linear system and also the best approach to solve it might vary. Therefore, careful selection of the matrix inversion approach is important to maintain computational efficiency (see Sec. 4.3).

Finally, our framework can shed light on when a potential tracker failure might occur. In general, the energy of the data term $\|\mathbf{A}_0\mathbf{w} - \mathbf{y}\|_2^2$ can serve as an indicator of how much the target representation changes from frame to frame. Intuitively, a drastic change in this energy from one frame to another might suggest that the tracker is drifting; however, this might not be the only reason for this change. For example, this energy can also fluctuate abruptly within a few frames due to illumination variation, deformation, occlusion, etc. Alone, the data term is not a reliable measure for target drift. However, in our formulation (Eq. (7)), the energy of the context term $\sum_{i=1}^{k} \|\mathbf{A}_i\mathbf{w}\|_2^2$ can be used to corroborate the implications of the data term. In many scenarios, an appearance change of the target does not affect the context (*e.g.* aspect ratio change, deformation, occlusion, etc.). Therefore, an abrupt change in both terms (data *and* context) within a few frames is a more reliable indication of tracking failure/drift. So, it is conceivable that monitoring for such events during tracking could help the tracker recover from an irreversible failure. For further discussion and results, refer to the **supplementary material**.

6. Conclusion

We propose a generic framework for correlation filter (CF) based trackers that incorporates context into the filter training stage at low computational cost. Extensive experiments show that our framework improves tracking performance for all tested CF trackers and is computationally efficient. While it improves performance across most attributes, we identify specific scenarios that benefit the most: fast motion, drastic appearance change (*e.g.* aspect ratio change or partial occlusion), and background clutter.

Acknowledgments. This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the VCC funding.

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[2] A. Bibi and B. Ghanem. Multi-template scale-adaptive kernelized correlation filters. In *IEEE International Conference on Computer Vision Workshops, ICCVW, 2015.*

[3] A. Bibi, M. Mueller, and B. Ghanem. *Target Response Adaptation for Correlation Filter Tracking*, pages 419–433. Springer International Publishing, Cham, 2016.

[4] D. S. Bolme, J. R. Beveridge, B. Draper, Y. M. Lui, et al. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010.*

[5] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[6] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *IEEE International Conference on Computer Vision, ICCV, 2015.*

[7] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1177–1184, June 2011.

[8] P. Dollar. Piotr's computer vision matlab toolbox: Histogram of oriented gradients.

[9] F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5135–5143. IEEE, 2015.

[10] 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, ECCV, 2012.*

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI, 2015.*, 2015.

[12] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 749–758, June 2015.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, Dec 2011.

[14] H. Kiani Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015.*

[15] H.-U. Kim, D.-Y. Lee, J.-Y. Sim, and C.-S. Kim. Sowp: Spatially ordered and weighted patch descriptor for visual tracking,. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[16] B. Kong and C. C. Fowlkes. Fast convolutional sparse coding (fcsc). *Department of Computer Science, University of California, Irvine, Tech. Rep*, 2014.

[17] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernandez, T. Vojir, G. Häger, G. Nebehay, R. Pflugfelder, et al. The visual object tracking vot2015 challenge results. In *Visual Object Tracking Workshop 2015 at ICCV2015*, Dec 2015.

[18] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision Workshops, ECCV, 2014.*

[19] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *Image Processing, IEEE . . .*, pages 1–14, 2015.

[20] S. Liu, T. Zhang, X. Cao, and C. Xu. Structural correlation filter for robust visual tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] C. Ma, J. B. Huang, X. Yang, and M. H. Yang. Hierarchical convolutional features for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082, Dec 2015.

[22] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5388–5396, 2015.

[23] M. Mueller, N. Smith, and B. Ghanem. *A Benchmark and Simulator for UAV Tracking*, pages 445–461. Springer International Publishing, Cham, 2016.

[24] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014.

[25] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[26] Y. Wu, J. Lim, and M.-H. Yang. Online Object Tracking: A Benchmark. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418. IEEE, June 2013.

[27] Y. Wu, J. Lim, and M. H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, Sept 2015.

[28] J. Xiao, L. Qiao, R. Stolkin, and A. Leonardis. *Distractor-Supported Single Target Tracking in Extremely Cluttered Scenes*, pages 121–136. Springer International Publishing, Cham, 2016.

[29] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

[30] T. Zhang, A. Bibi, and B. Ghanem. In defense of sparse tracking: Circulant sparse tracker. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.*

[31] G. Zhu, J. Wang, Y. Wu, and H. Lu. Collaborative correlation tracking. In X. Xie, M. W. Jones, and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 184.1–184.12. BMVA Press, September 2015.