When Correlation Filters Meet Siamese Networks for Real-time Complementary Tracking

Dongdong Li, Fatih Porikli, *Fellow, IEEE,* Gongjian Wen, *Member, IEEE* and Yangliu Kuai

***Abstract*—Discriminative Correlation filter (DCF) based trackers have recently exhibited high efficiency and impressive robustness to challenging factors such as illumination change and partial occlusion. However, in cases with fast motion and full occlusion, these trackers drift off soon and can hardly re-detect the target from the restricted search region due to the boundary effect. On the contrary, recent work using a fully-convolutional Siamese network (Siamfc) locates the exemplar image within a large search image but suffers from coarse location and distractors. In this paper, we propose a Real-time Complementary Tracker (RCT) by integrating DCF and Siamfc into a two-stage tracking framework where DCF and Siamfc share mutual advantages and complement each other. In the first stage of this framework, RCT locates the target coarsely but robustly with Siamfc. In the second stage, the derived coarse location is refined by DCF for higher accuracy. For efficiency reasons, Siamfc in the first stage is activated occasionally based on the tracking status inferred from the correlation response map of DCF in the second stage. Comprehensive experiments are performed on three popular benchmark datasets: OTB2013, OTB2015 and VOT2016. On OTB2013, RCT runs with over 40 fps and achieves an absolute gain of 4.8% and 5.2% in mean overlap precision compared with two base trackers (Staple and Siamfc). On VOT2016, RCT makes a good balance between performance and efficiency, ranking fifth in EAO and first in EFO compared with the top 5 trackers.**

***Index Terms*—correlation filter, Siamese network, complementary tracking.**

I. Introduction

VISUAL object tracking is an established yet rapidly evolving research area in computer vision. In general, it aims to estimate the spatial trajectory of a target object in an image sequence, given its initial state, i.e. location and underlying area. It provides a fundamental component for high-level visual understanding problems such as motion analysis, event detection, situational awareness, and activity recognition. Despite significant progress in recent years, finding the corresponding object regions across multiple frames is still a challenging problem due to factors such as occlusion, deformation, illumination change, fast motion and background clutter. In this paper, we focus on single-camera, single-target, short-term and model-free tracking, and refer readers to [1], [2], [3] for a thorough overview of existing algorithms.

Most tracking approaches adopt the tracking-by-detection principle to locate the target object. According to their consolidated appearance representation schemes, tracking-by-detection methods can be categorized into generative methods [4], [5], [6] and discriminative methods [7], [8], [9]. Generative methods model target appearance often by ignoring the background information, which leads to drift issues in complex scenes, while discriminative methods pose tracking task as a binary classification objective that discriminates the object from its surrounding background. Recently, correlation filter based discriminative trackers demonstrated significant performance improvement in terms of accuracy and robustness on several benchmarks [3], [2]. Generally speaking, correlation filters locate the target object in a very restricted target search area. This limitation is due to the fact that the detection scores are accurate only near the center of the region as a result of the boundary effects. This confined search region often causes the estimated region to stay behind the target object, contaminating the object model with background information, as in Figure 1 in the presence of fast motion and full occlusions. SRDCF [10] alleviates this boundary effect to some extent by expanding the search area to a more extensive region with a spatial regularization component. However, such an expansion of the search region comes at the price of a significant slowdown in the tracking speed.

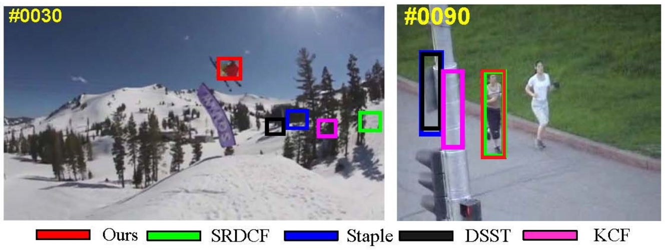


Fig. 1: Comparisons of our approach with four other state-of-the-art correlation filter based trackers in challenging scenarios of fast motion on *skiing* sequence (left) and full occlusion on *jogging-1* sequence (right) in the OTB2013 benchmark dataset [1]. As visible, our approach (red) was able to catch up with the fast-moving skier in *skiing* and re-detect the person after the full occlusion in *jogging-1*.

Recent works [11], [12], [13] have shown impressive performance using Siamese networks. Among them, Siamfc [11] addresses visual tracking as a universal similarity learning problem with an offline trained Siamese network. Without model updating, Siamfc achieves state-of-the-art performance and runs with over 80 fps on Graphics Processing Units (GPUs). On the one hand, the absence of model updating avoids fine-tuning during online tracking, yet on the other, it blurs the intra-class difference. As a result, Siamfc often drifts

D. Li, G. Wen and Y. Kuai are with College of Electronic Science, National University of Defense Technology, Changsha, Hunan, China (e-mail: moqimubai@sina.cn).

F. Porikli is with Australian National University.

to similar background regions (hard negatives) in complex scenarios as shown in Figure 2.

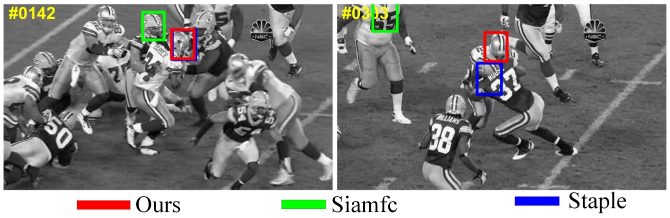


Fig. 2: Results of our approach, Siamfc and Staple on *football* sequence where the scenes contains similar objects, from the OTB2013 benchmark dataset [1]. Both Siamfc and Staple fail to keep the original target and drift towards other similar objects; the offline Siamese network generates high scores for any region that has a similar appearance to the initial target appearance while Staple is stranded in regions that have similar color cues. In comparison, our approach is resistant to such distractors.

Despite different tracking mechanisms, correlation filters and Siamfc are mutually complementary in the following ways:

(1) *Online* versus *Offline*. Correlation filters perform model update in an interpolated manner while Siamfc is initialized once with the bounding box in the first frame.

(2) *Spatial* versus *Semantic*. Correlation filters usually employ handcrafted features (*e.g.,* HOG [14]) which are sensitive to pose, spatial arrangement of parts, and local texture. In contrast to correlation filters, Siamfc extracts features maps from the higher layers of a deep neural network. These features maps are robust to significant appearance variations (depending on the training process and data augmentation) compared with the handcrafted features.

(3) *Fine-grained* versus *Coarse-grained*. In comparison to low-level handcrafted features, the semantically interpretable feature map in Siamfc is relatively coarse for accurate location.

(4) *Small Search Region* versus *Large Search Region*. Correlation filters can only locate the target in a limited search region due to the boundary effects. However, Siamfc can execute search in a larger region to cope well with fast object/camera motion and full occlusions.

Inspired by the aforementioned observations, we propose a real-time running complementary tracker (termed as RCT) that takes advantage of the merits of correlation filters and Siamfc. In each frame, RCT estimates the target translation in a two-stage scheme. In the first stage, RCT locates the target coarsely but robustly with semantic convolutional features in a larger search region. The semantic feature maps of Siamfc are extracted as the activations from the fifth convolutional layer. Therefore, these feature maps are in low-resolution yet representative of the high-level semantics, which are considered to be robust to significant appearance variations of the target. In the second stage, the coarse location derived in the first stage is refined by a correlation filter for higher spatial resolution positioning of the target object. For efficiency purposes, Siamfc is activated infrequently when the peakiness of the response map of the correlation filter is below an adaptive threshold. Rather than searching jointly in both translation and scale dimensions, we follow Danelljan *et al*. [15] and learn a distinct, multi-scale template for scale search using a 1D Correlation Filter.

We perform an extensive set of experiments on three popular benchmark datasets: OTB2013 [1], OTB2015 [2] and VOT2016 [3]. Compared with a single correlation filter or a Siamese network based tracker, our RCT method consistently outperforms the baseline trackers on OTB2013 in the success plots using the area under the curves (AUC) while maintaining an average tracking speed of over 40 fps. On VOT2016, RCT makes a good tradeoff between the accuracy and efficiency, ranking fifth in EAO and first in EFO among the top 5 trackers.

II. Related Work

For completeness, we provide a brief overview of the most relevant works.

Correlation Filter for online Tracking. In recent years, correlation filter based trackers have shown continuous performance improvements in terms of accuracy and robustness. Standard correlation filters transform spatial correlation into element-wise multiplication in the Fourier domain and attract considerable attention in the tracking community due to their extremely high computational efficiency. Different variants of correlation filters have been proposed to boost tracking performance using multi-dimensional features [16], robust scale estimation [15], non-linear kernels [9], long-term memory components [17], target response adaptation [18] and complementary cues [19]. The pioneering MOSSE tracker proposed by Bolme *et al.* [20] conducts spatial correlation in the frequency domain and achieves a runtime of over 600 frames per second (fps). Later, correlation filters have been extended to multi-dimensional features for visual tracking. Seminal followup work by Henriques *et al*. [9] formulated learning correlation filters as a ridge regression problem and exploited circular correlation at both learning and detection stages. A discriminative scale space tracker (DSST) [15] is proposed by Danelljan *et al.* to achieve real-time scale adaptive tracking. Bertinetto *et al.*[19] combine a correlation filter and a global color histogram to achieve robustness to both deformation and color change.

Despite the above mentioned achievements, two issues remain unsolved in correlation filter based real-time tracking:

(a) The restricted search region. Standard correlation filter based trackers suffer from period assumption induced by circular correlation. This leads to a restricted search region because the correlation scores are only accurate near the center of the search region. SRDCF [10] alleviates the boundary effect and expands the search area to a larger region. However, the real-time tracking speed is sacrificed at the same time due to the Gauss-Seidel iterations in the spatially reguralized component. Therefore, how to extend the search region while maintaining a real-time speed remains an unsolved issue for correlation filters.

(b) Linear interpolation for model update. Generally, correlation filter based trackers linearly interpolate the filter coefficients in the previous and current frames to cope with target appearance variation. However, this linear interpolation heavily relies on the spatio-temporal consistency of visual cues and suffers from noisy model update in presence of abrupt motion, severe deformation and heavy occlusion. Therefore, how to adapt to appearance change while maintaining model stability (*i.e.* stability-plasticity dilemma) remains a pending problem for correlation filter based tracking.

Siamese Architecture for Tracking. Recently, Convolutional Neural Networks (CNNs) have significantly advanced the state-of-the-art in many vision applications, such as image classification [21], object detection [22] and saliency detection [23]. Driven by this popular trend, many correlation filter based trackers [24], [25] substitute handcraft features with deep convolutional features and achieve superior tracking performance. However, simply regarding CNN as a feature extractor does not take full advantage of the benefits of end-to-end learning. To fully exploit the representation power of CNN in visual tracking, it is desirable to train them on large-scale dataset specialized for visual tracking. Prior works [26], [27] train CNN offline with massive data and perform SGD (stochastic gradient descent) to fine-tune multiple layers of the network during online tracking. All these methods achieve state-of-the-art results but fail to operate in real-time. Recently, the Siamese architecture has been exploited in the tracking field and shows impressive performance. Tao *et al*. [12] propose to train a Siamese network to identify candidate image locations that match the initial object appearance and term their method as Siamese INstance search Tracker (SINT). Held *et al*. [13] introduce GOTURN which avoids the need to score many candidate patches and runs at 100 fps. However, GOTURN does not possess intrinsic invariance to translation of the search image. Later, Luca *et al*. train a similar Siamese network (Siamfc) to locate an exemplar image within a large search image. The network parameters are trained from scratch on the ILSVRC Imagenet Video dataset [28]. Despite its extreme simplicity and absence of model updating, Siamfc achieves state-of-the-art performance on multiple benchmarks. In this section, we argue that there are three factors which may hinder the applicability of Siamfc:

(a) Low efficiency without GPU. Siamfc only runs in real-time on GPU due to the convolution operations in the Siamese networks. The high computational expense limits its applicability in many real-time applications with limited hardware capabilities, such as aerial tracking using unmanned aerial vehicle (UAV).

(b) The low-resolution score map. Siamfc extracts high-level semantic feature maps from both the exemplar image and the search image. A low-resolution score map is obtained by computing cross-correlation of the two feature maps on a 17×17 grid. Although tricks like bicubic interpolation are applied to up-sample the score map, the score map is still relatively coarse for fine-grained tracking.

(c) Confusion with distractors. Without model updating, Siamfc is free from noisy update but, on the other hand, is blind to target appearance variation. Meanwhile, the high-level semantic features fail to capture the spatial details of the target appearance. As a result, Siamfc ignores the intra-class variation and tends to drift to similar confusing objects resembling the initial target appearance.

Combining Correlation Filters and CNNs Discriminative Correlation Filter (DCF) and Convolutional Neural Network (CNN) based trackers have achieved considerable popularity in the tracking community. This phenomenon is especially evident from the outcome of the Visual Object Tracking (VOT) 2016 challenge [3], where eight trackers are based on either DCF or CNN among the top ten trackers. Therefore, we argue that state-of-the-art tracking performance can be further improved if we establish a unified tracking framework utilizing the advantages of both DCF and CNN. By now, only limited work focuses on the combination of DCF and CNN due to their different tracking mechanism. Danelljan *et al.* [24] conduct the pioneering work by introducing deep convolutional features into the DCF-based tracking framework. Ma *et al.* [29] learn correlation filters on each convolutional layer to encode the target appearance and hierarchically inter the maximum response of each layer to locate the targets. Recently, Ma *et al.* [30] investigate the potential of correlation filters as the counterparts of convolution filters in deep neural networks for tracking. Inspired by [29], [19], in this work, we propose a hybrid tracking framework in which DCF and CNN share their mutual advantages and complement each other.

III. Building Blocks

We employ *Staple* [19] and *Siamfc* [11] as the building blocks for our tracker RCT. It is worth to mention that RCT is a very flexible tracking framework and our implementation is far from optimal. We believe there is a room for future improvement and generalization. In the following discussion, we give a brief overview of two building blocks incorporated in our tracker.

A. Siamfc

Compared with correlation filter based trackers, the advantage of Siamfc is that, instead of a candidate image of the same size, we can provide as input to the network a much larger search image and it will compute the similarity at all translated sub-windows on a dense grid in a single evaluation. This larger search image equips Siamfc with a large field of vision to cope with abrupt motion and heavy occlusion.

Siamfc applies an identical transformation φ (similar to [31]) to both the exemplar image z and search image x and combines the resulting feature maps using a cross-correlation layer

$$f(z,x) = \varphi(z) * \varphi(x) + b\mathbb{1} \tag{1}$$

where $b\mathbb{1}$ is a vector which takes the value $b \in \mathbb{R}$ in all of its coefficients, and $f(z,x)$ is the real-valued score of a single exemplar-candidate pair.

The neural network is trained on both positive and negative pairs adopting the logistic loss

$$\ell(y, f(z,x)) = log(1 + exp(-yf(z,x))) \tag{2}$$

where $y \in \{+1, -1\}$ is the ground-truth label of the exemplar-candidate pair.

Since the search image is larger than the exemplar image, Siamfc generates a score map D where the loss is defined as the mean of the individual losses as described as following,

$$L(y, f) = \frac{1}{|D|} \sum_{u \in D} l(y[u], f[u]). \tag{3}$$

The parameters θ of φ are obtained by applying Stochastic Gradient Descent (SGD) to the problem

$$\arg\max_{\theta} \mathbb{E}_{(z,x,y)} L(y, f(z, x; \theta)). \tag{4}$$

During the tracking process, a larger image patch is cropped and fed into the Siamese network, which generates the response map for target localization. Readers can refer to [26] for other details in training data preparation and network training.

B. Staple

Combining a correlation filter (using HOG features) with a global color histogram, *Staple* solves two independent ridge-regression problems, exploiting the inherent structure of each representation. Here, we briefly describe the Staple formulation, adopting the same notations as in [19] for convenience. The score function of *Staple* is a linear combination of template and histogram scores:

$$f(x) = \gamma \cdot f_{tmpl}(x) + (1 - \gamma) \cdot f_{hist}(x) \tag{5}$$

where γ is an interpolation parameter, f_{tmpl} is the template score and f_{hist} is the histogram score.

The aim of the template model is to learn a d-dimensional correlation filter h from a d-dimensional feature f. We denote the feature layer $l \in \{1, \cdots, d\}$ of f by f^l. The desired output of y is a scalar valued function, which includes a label for each location in the feature f. The desired correlation filter h is obtained by minimizing the following target function,

$$\varepsilon(h) = \left\| \sum_{l=1}^{d} f^l * h^l - y \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2. \tag{6}$$

Here, $*$ denotes the convolution operator and the regularization scalar λ controls the impact of the regularization term.

Based on the circulant assumption, the solution to (6) is derived as following

$$\hat{h}^l = \frac{\hat{f}^{l^*} \cdot \hat{y}^l}{\sum_{l=1}^{d} \hat{f}^{l^*} \cdot \hat{f}^l + \lambda}. \tag{7}$$

Here, $\hat{f}_j^l$ means the Fourier transform of f_j^l and $\hat{f}_j^{l^*}$ means the complex conjugation of $\hat{f}_j^l$. The product and division in (7) is point-wise.

By contrast, we calculate the object likelihood of each pixel belonging to the foreground object in the histogram model. Let $H_{\Omega}^{I}(b)$ denote the frequency of the b-th bin of the color histogram H computed over the region $\Omega \in I$. Given an image patch I centered at the target, the normalized object histogram H_O^I and background histogram H_B^I can be derived from the target area and the surrounding background area, respectively. The object likelihood of a given pixel x in I can be obtained as

$$P(x \in O | I, x) = \frac{P(x \in O|I)}{P(x \in O|I) + P(x \in B|I)}. \tag{8}$$

Here, the foreground and background likelihood can be directly estimated from color histograms, *i.e.* $P(x \in O|I) = H_O^I(b_x)$ and $P(x \in B|I) = H_B^I(b_x)$, where b_x denotes the color bin b assigned to the pixel x.

Hence,

$$f_{hist}(p) = \sum_{x} P(x \in O), \tag{9}$$

where x is the pixel in the bounding box centered at p. The histogram score function can be efficiently evaluated using the integral histogram [32].

IV. Proposed Real-time Complementary Tracker

RCT seeks an efficient solution to integrate Siamfc and Staple. Its components are described in the following subsections.

A. Coarse Translation Initialization

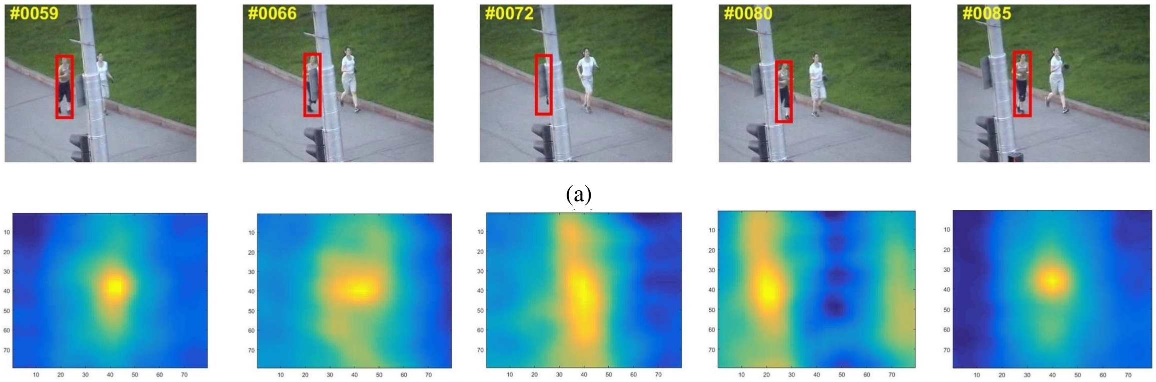
Different from correlation filters which suffer from the restricted search region, Siamfc holds a large search region, almost four times the target size. The large search region enables Siamfc to cope better with fast motion and heavy occlusion than correlation filters as shown in Figure 1.

Besides, as described in Section III-A, Siamfc extracts deep feature maps from both the exemplar and search images and then computes the score map from cross-correlation of the two maps. Compared with correlation filters using handcrafted features (*e.g.,* HOG), the deep feature maps in Siamfc are more effective to encode the semantic appearance variations. Therefore, together with the global color histogram in Staple, Siamfc greatly improves the robustness of RCT in terms of severe deformation, illumination change and background clutter.

B. Refined Translation Estimation

It has been demonstrated that [24], unlike image classification, the shallow layers achieve better tracking performance than deeper layers. This effect is partly attributed to the decreasing spatial resolution from the first layers to the last layers. The score map derived by Siamfc is in low resolution (17×17) due to strides in the embedding network. As a result, Siamfc is insufficient for capturing fine-grained spatial detail which is important for accurate location.

On the other hand, correlation filters using low-level handcrafted features (*e.g.,* HOG) retain more fine-grained spatial details and thus are useful for precise localization. In light of this observation, a coarse-to-fine searching strategy is adopted in RCT. We propose to integrate Siamfc and Staple for translation estimation, where both semantics and fine-grained details are simultaneously exploited to handle large appearance variations and sampling ambiguity. RCT first locates the target coarsely but robustly with Siamfc. Subsequently, the coarse location is refined by Staple for higher accuracy. Meahwhile, equipping offline Siamfc with online Staple also helps RCT



(a)

(b)

Fig. 3: (a) Screenshots of tracking results of *RCT* on the *Jogging-1* sequence. (b) Corresponding response maps of *RCT*. In frame 72, the target undergoes heavy occlusion and thus the energy in the derived response map is dispersive compared with that in the other frames.

adapt to the latest target appearance and thus alleviate confusion with distractors as shown in Figure 2.

Therefore, it's necessary to initialize Staple with the target location of Siamfc to expand the search region and refine location accuracy as well.

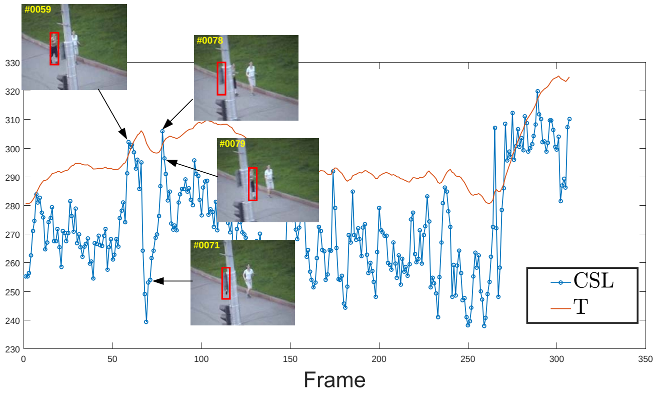


Fig. 4: The values of CSL and adaptive threshold on all frames of the *jogging-1* sequence. In frame 59, the target begins to undergo heavy occlusion. In frame 78, the CSL value is above the adaptive threshold T and thus the *Siamfc* component in *RCT* is activated. As a result, the target is re-detected in frame 79.

C. Scale Estimation

In the source codes provided in [11], scale variation is estimated by processing the search image at several scales with the a fixed aspect ratio. With no doubt, searching scale at multiple resolutions significantly increases the computational cost. To achieve real-time scale adaptive tracking, RCT removes the scale estimation from Siamfc. Instead, Staple and Siamfc shares a common 1-dimensional scale correlation filter after the two-stage translation estimation process.

D. Tracker Switch

Most frames in a video are ‘easy’ frames where the target moves smoothly and its appearance changes slowly. By contrast, ‘hard’ frames appear only occasionally when the target undergoes partial occlusion, fast motion or significant appearance variation. In ’easy’ frames, Staple locates the target more accurately and efficiently than iamfc because Staple has a smaller feature stride. However, in ’hard’ frames, Siamfc is more robust to strong appearance change than Staple. This is because Siamfc is trained with massive data in an end-to-end manner while Staple is trained with only limited online data.

Based on this observation, we automatically activate Siamfc with a tracking switch instead of tracking with Siamfc and Staple in each frame. In most frames of a given video, RCT switches Siamfc off to track in real-time. In occasional challenging frames, RCT switches Siamfc on to track in a coarse-to-fine manner as described in Sections IV-A and IV-B, and avoids potential tracking failures. Whether switching Siamfc on-or-off depends on the tracking status of RCT.

We point out that the tracking status of RCT can be inferred from the peakiness of the response map of Staple. In the general case, this response map peaks at the highest and damps fast from the peak to the boundary. However, in presence of heavy occlusion, deformation and abrupt motion, the response map damps slowly due to the sidelobe leakage as shown in Figure 3. In light of this observation, we design the centralized sidelobe leakage (CSL) measure to quantify the peakiness of the response peak and thus evaluate the tracking status. The lower CSL value means the more reliable tracking result and thus the better tracking status.

Here we describe the CSL measure in detail. Let f_t be a $N \times M$ matrix representing the response map of frame t. The location of the response peak is given as

$$[\mu, \nu]^T = \arg\max_{i,j} f_t^{i,j}. \tag{10}$$

where $f_t^{i,j}$ corresponds to the j_{th} element of the i_{th} row of f_t and $[\mu, \nu]$ stand for the pixel coordinates of the maximum on the response map.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2892759, IEEE Transactions on Circuits and Systems for Video Technology

The CSL measure of the response map f_t is defined as

$$CSL = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} |i-\mu| \cdot |j-\nu| \cdot f_t^{i,j}}{\sum_{i=1}^{N}\sum_{j=1}^{M} f_t^{i,j}} \tag{11}$$

where the denominator provides a normalization based on the amplitude of each pixel on the response map.

In this paper, we assume that a surge of CSL occurs when the target encounters challenging situations such as occlusion or fast motion. To robustly detect the surge in CSL, we damp the fluctuations of CSL with a forgetting rate λ and derive an adaptive threshold T as

$$\begin{cases} CSL'_0 = CSL_1, \\ CSL'_t = \lambda \cdot CSL'_{t-1} + (1-\lambda) \cdot CSL_t, \\ T = \eta \cdot CSL'_t, \end{cases} \tag{12}$$

where λ is an interpolating parameter, $\eta > 1$ is an amplified parameter, CSL_t is the CSL measure of f_t and CSL'_t is a smoothed version of CSL_t.

During tracking, η is fixed while T is updated in each frame as Equation 12. To robustly detect big fluctuation of CSL, the interpolated version of CSL, namely CSL', is computed to smooth small fluctuation. In RCT, the Siamfc component is switched on if the CSL value is above T, which indicates unreliable tracking result and bad tracking status of RCT in the previous frame. Figure 4 shows the corresponding values of CSL and T of RCT over all frames in the *Jogging-1* sequence. As shown in 4, there exists a significant surge in the line of CSL around frame 59. The CSL value in frame 59 is above the smooth line of the threshold T.

E. Overall Tracking Framework

Here, we present an outline of RCF in Algorithm 1 and show the diagram of RCF in Figure 5.

Algorithm 1 Real-time Complementary Tracking

```
Input:
    Target state $X_{t-1} = (x_{t-1}, y_{t-1}, s_{t-1})$ in frame $t-1$.
Output:
    Estimated target state $X_t = (x_t, y_t, s_t)$ in each frame.
if $mode$ = 1 then
    Estimate the target location $(x_t, y_t)$ with Siamfc around $(x_{t-1}, y_{t-1})$ in frame $t$. Refine $(x_t, y_t)$ by searching around $(x_t, y_t)$ with Staple.
else
    Estimate the target location $(x_t, y_t)$ with Staple around $(x_{t-1}, y_{t-1})$ in frame $t$.
Estimate the target scale $s_t$ with DSST.
Calculate the CSL value from the response map of Staple as in Equation 11.
Update the adaptive threshold $T$ as in Equation 12.
if CSL $>$ $T$ then
    $mode$ = 1
else
    $mode$ = 0
Update the correlation filter and color histogram in Staple and the 1-dimensional correlation filter in DSST.
```

V. Experiments

Here, we present a comprehensive evaluation of the proposed tracker (RCT). Results are reported on three benchmark datasets: OTB2013, OTB1002015 and VOT2016.

A. Details and Parameters

For all the experiments in this paper, we follow the same parameter setting of *Staple* and *Siamfc* as reported in [19] and [11] respectively. In this subsection, we mainly detail the parameter setting related with the CSL measure. The forgetting rate parameter λ in Equation 12 is set to 0.95 and the amplifying factor η in the adaptive threshold T is set to 1.1. All the parameters are fixed for all videos and datasets. RCT is implemented in Matlab with MatConvNet[33] and runs on a desktop computer with a core Intel Core i5-5200 CPU at 2.2 GHz. The source codes and experimental results are available at https://github.com/moqimubai/Realtime-Complementary-Tracking.

B. Baseline Comparison

Here, we compare the proposed tracker RCT with three trackers (Siamfc, Staple and RCT$_{ws}$) on the OTB2013 benchmark. Siamfc and Staple are two baseline trackers while RCT$_{ws}$ stands for a tracker similar to RCT but without the tracker switch. In other words, both Siamfc and Staple in RCT$_{ws}$ are always used in each frame in a coarse-to-fine manner.

TABLE I: A comparison of RCT with baseline trackers on OTB2013

	RCT	RCT$_{ws}$	Staple[19]	Siamfc[11]
Mean OP (%)	74.26	65.98	69.17	63.90

Table I showns the mean overlap precision (OP) for four methods (RCT, RCT$_{ws}$, Staple, Siamfc) on the OTB2013 dataset. OP is computed as the fraction of frames in the sequence where the intersection-over-union overlap with the ground truth exceeds a threshold of 0.5. As shown in Table I, Staple and Siamfc achieve a mean OP of 69.12% and 63.90% respectively. RCT$_{ws}$, without the tracker switch, makes a compromise between *Staple* and *Siamfc* and achieves a mean OP of 65.98%. This is because that the coarse location provided by Siamfc is so far off the location identified by Staple that Siamfc distract Staple in during tracking. On contrast, with the tracker switch, our RCT outperforms both baseline trackers and achieves a mean OP of 74.26%. We owe the superior performance of RCT to the elegant combination of Staple and Siamfc with the automatic tracker switch. With the tracker switch, RCT takes advantages of the high location accuracy of Staple and the larget search area of Siamfc at the same time.Thats the reason why RCT achieves better performance than RCT$_{sw}$.

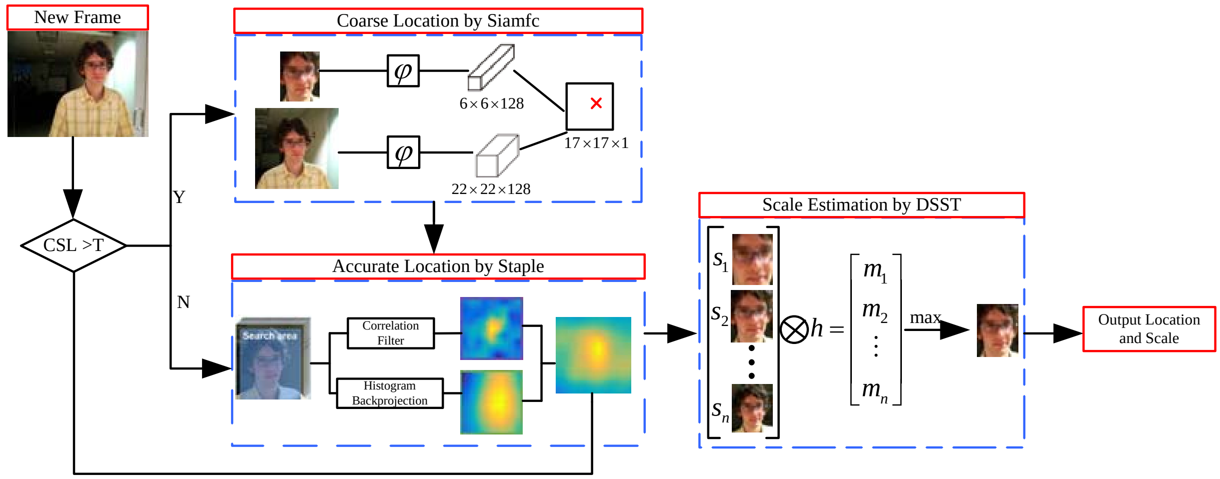
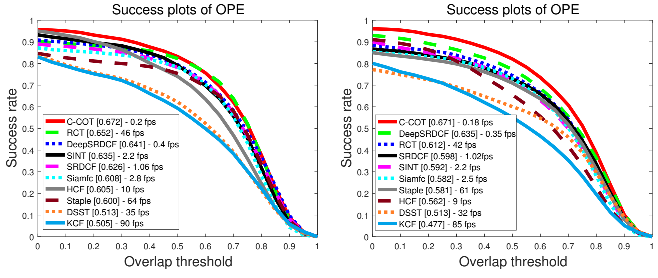
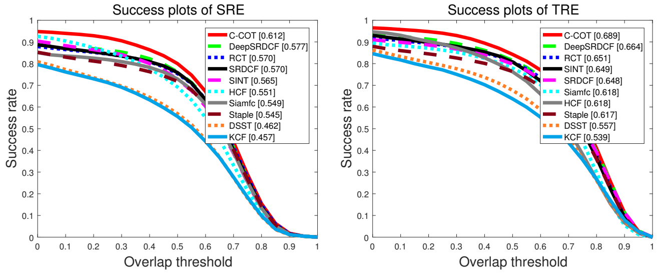


Fig. 5: Main components of the proposed real-time complementary tracking algorithm.



(a) (b)

Fig. 6: Success plots showing a comparison with state-of-the-art methods on OTB2013 (left) and OTB-2015 (right) benchmark datasets. Our RCT achieves an average speed of over 40 fps and outperforms all trackers except C-COT and DeepSRDCF which run slower than 1 fps.



(a) (b)

Fig. 7: Comparison with respect to robustness to initialization on OTB2013. We show success plots for both the spatial (SRE) and temporal (TRE) robustness.

C. OTB Benchmark

We provide a comparison of RCT with 9 trackers from the literature: C-COT[25], SRDCF[10], Staple[19], DSST[15], KCF[9], Siamfc[11], SINT[12], DeepSRDCF[24] and HCF[29]. Among these trackers, C-COT[25], SRDCF[10], Staple[19], DSST[15] and KCF[9] are based on correlation filters. Siamfc[11] and SINT[12] are based on Siamese networks. DeepSRDCF[24] and HCF[29] are based on both correlation filters and deep learning.

1) State-of-the-art Comparison: Figure 6 shows the success plots on the OTB2013 and OTB2015 benchmark datasets. The success plot shows the mean overlap precision (OP), plotted over the range of intersection-over-union thresholds. The trackers are ranked using the *area under the curve* (AUC), displayed in the legend. On OTB2013, RCT runs with 46 fps and provides the second best performance, with an AUC score of 65.2%. Our approach obtains a significant gain of 5.2% and 4.4% in AUC score compared with Staple and Siamfc. It's worth mentioning that, among the ten trackers, only RCT and Staple achieve state-of-the-art performance and real-time speed without GPU.

2) Attribute Based Comparison: We perform an attribute based analysis of RCT on the OTB2013 dataset. All the videos in OTB2013 are annotated with 11 different attributes, namely: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low resolution. Despite no explicit deformation or occlusion handling component, our tracker performs favorably in cases with deformation and occlusion as shown in 9.

3) Qualitative Comparison: Due to page limitation, we compare RCT with other four trackers (Siamfc[11], Staple[19], DSST[34], KCF[9]) on 11 challenging sequences in Figure 10. Staple performs well in the most sequences. However, it drifts due to fast motion in sequence *skiing* and occlusion in sequence *jogging-1* and *girl*. Siamfc is robust to a number of challenging situations like occlusion, abrupt motion and scale change. However, it is sensitive to poor illumination in sequence carDark and confusing scenes in sequences *bolt* and *football*.

It's worth noting that the tracking mechanism of RCT copes well with fast motion, distractors and partial occlusion. However, like most trackers, RCT drifts to the background in presence of long-term and/or full occlusion. Figure 8 shows tracking failures of RCT due to full occlusion. In future works, we tend to equip RCT with a re-detection [17] and/or re-identification[35], [36] module to achieve long-term tracking.

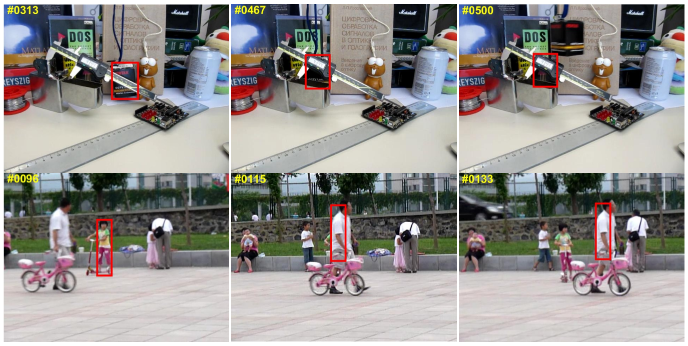


Fig. 8: Screenshots of tracking failures for RCT over two different videos undergoing long-term and full occlusion. The videos from top to bottom are *Box* and *Girl2* from the OTB2015 dataset.

D. VOT2016

The visual object tracking (VOT) challenge is a competition between short-term, model-free visual tracking algorithms. Different from OTB, for each sequence in this dataset, a tracker is restarted whenever the target is lost (*i.e.* at a tracking failure). Four primary measures are used to analyze tracking performance: accuracy (A), robustness (R), expected average overlap (EAO) and equivalent filter operation (EFO). A is calculated as the average Intersection-over-Union (IoU), while R is expressed in terms of the total number of failures. EAO represents the average IoU with no re-initialization following a failure. EFO reports the tracker speed in terms of a predefined filtering operation that the toolkit carries out prior to running the experiments. For our experiments, we use the latest stable version of the VOT toolkit (*i.e.* VOT2016 toolkit). We refer to [3] for details.

Table II shows the comparison of our approach with the top 5 participants in the VOT2016 challenge. Figure 11 shows a visualization of the overall results on the VOT2016 dataset. In the comparison, RCT ranks fifth in terms of EAO and ranks first in terms of EFO. Among the top five trackers (ranked by EAO), only RCT achieves the real-time speed almost 22-fold speed up in EFO compared to the top tracker C-COT. Despite its simplicity, our RCT improves over recent state-of-the-art real-time trackers (Figures 6 and 11). RCT outperforms most of the best methods in the VOT2016 benchmark while maintaining high frame-rate speed (Figure 6 and Table II).

E. Detailed analysis of RCT

Different base trackers. Our RCT tracker is composed of two basic cooperative components, the Siamese component, and the DCF component. To show the effects of different basic components, we perform two groups of comparison experiments on OTB2013. The mean overlap precision and average tracking speed is provided in Table III and Table IV. We compare two different choices of the Siamese component including Siamfc[11] and SINT[12] in Table III. It shows that Siamfc (74.26% and 46 fps) is slightly less accurate but more efficient than SINT (75.58% and 40 fps).

TABLE II: State-of-the-art in terms of expected average overlap (EAO), robustness (failure rate), accuracy, and speed (in EFO units) on the VOT 2016 dataset. Only the top-5 best compared trackers are shown. The best and second best values are highlighted by red and blue fonts.

	Staple	**RCT**	MLDF	SSAT	TCNN	C-COT
EAO	0.295	0.299	0.310	0.320	0.325	0.331
Failure rate	1.35	1.37	0.83	1.04	0.96	0.85
Accuracy	0.544	0.560	0.490	0.577	0.554	0.539
EFO	11.144	11.63	1.483	0.475	1.049	0.507

We further compare three different choices of the DCF component including Staple [19], DSST[34] and KCF[9] in Table IV. Our results show that KCF (60.48 %) is less accurate than DSST (65.18 %) and Staple (74.26 %) due to the absence of the scale estimation module. Meanwhile, Staple (74.26%) performs better than DSST (65.18%) due to the additional color histogram which is robust to target deformation. Though DSST runs faster than Staple (given the same cell size for the HOG[14] feature), RCT with DSST (38 fps) runs more slowly than RCT with Staple (46 fps). This is a result of the fact that DSST is less accurate and robust than Staple, which results in more frequent activation for the Siamese component, hence increasing the computation.

Different peakiness criterion. The tracking status of our RCT tracker is inferred from the peakiness of the response map of the DCF component. Therefore, a good peakiness criterion can effectively switch Siamfc on in hard frames to avoid tracking drift and switch Siamfc off in easy frames to avoid extra computation. To show the effects of different peakiness criterions, we compare two different choices including our centralized sidelobe leakage (CSL) and the peak-versus-noise ratio (PNR) [37]. The CSL value measures the peakiness of the response map based on each value on the response map. Nevertheless, the PNR value measures the peakiness only based on the maximum and minimum values on the response map.

The comparison of the mean overlap precision and tracking speed on OTB2013 is provided in Table V. It shows that RCT with CSL achieves higher accuracy and lower efficiency than RCT with PNR. This is because CSL obtains more information from the response map than PNR, which results in slightly more frequent activation of Siamfc.

TABLE III: Comparison of different Siamese components in RCT with Staple as the DCF component.

Tracker	RCT with Siamfc	RCT with SINT
Mean OP (%)	74.26	75.58
Avg. FPS	46	40

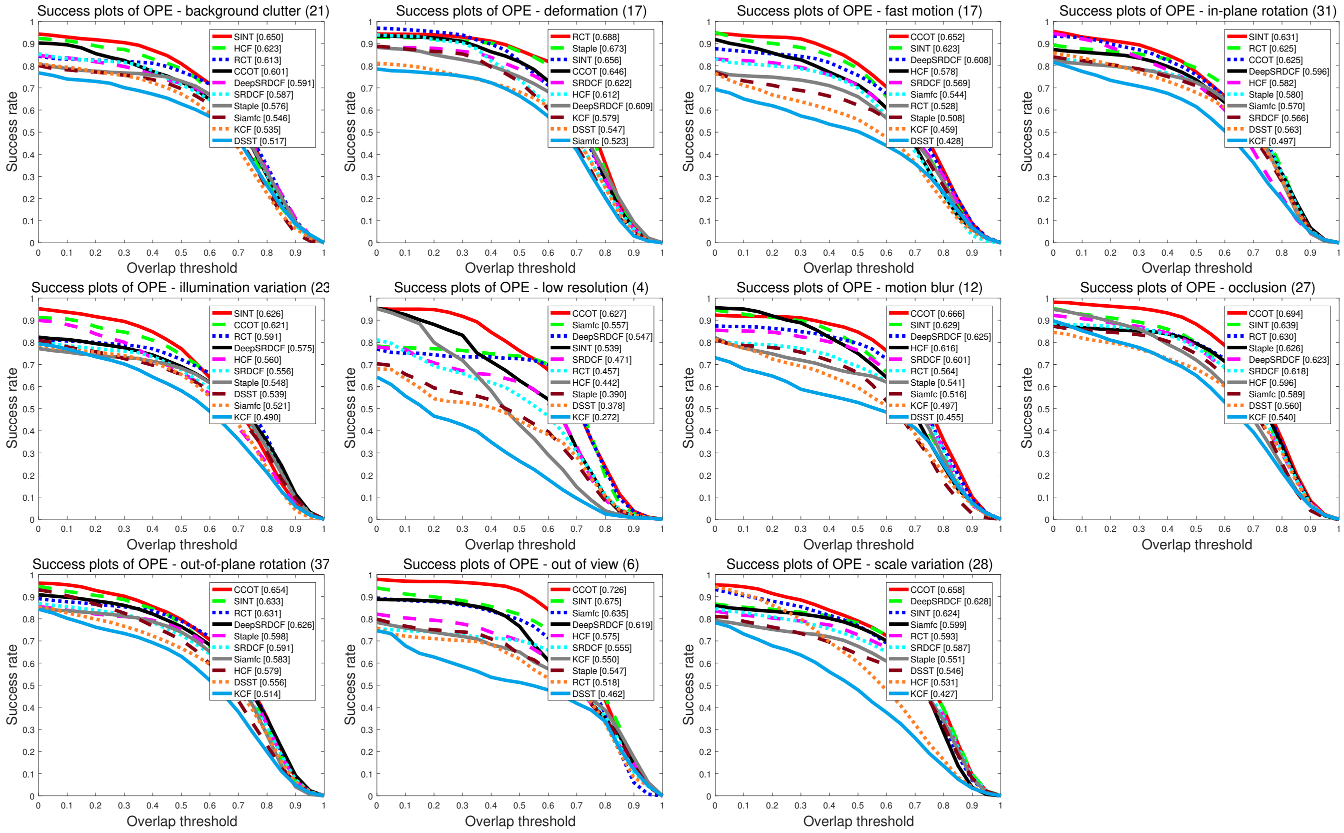


Fig. 9: *Success ratio* plots on 11 attributes of the OTB-2013 dataset. Trackers are ranked by their AUC scores. Ours method has achieved consistently the superior performance over the state-of-the-art.

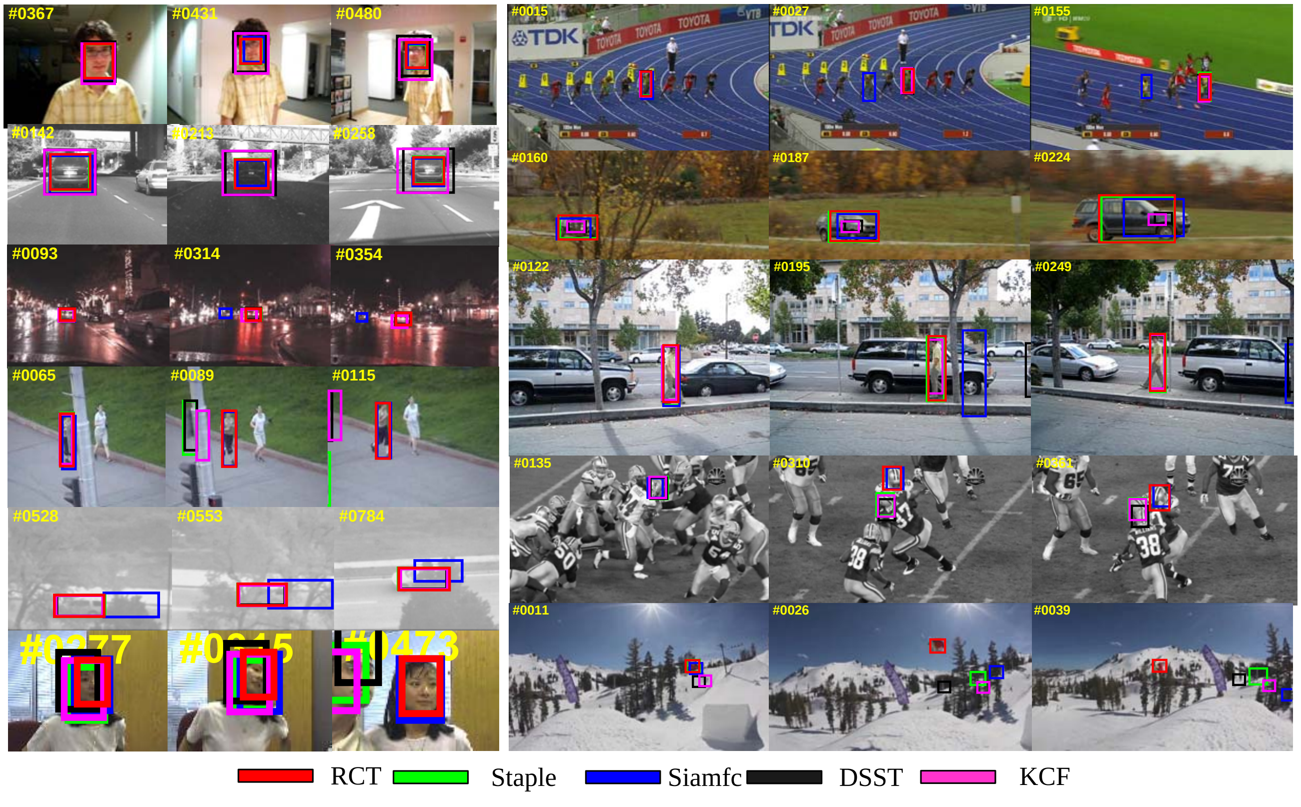


Fig. 10: Snapshots of RCT and the compared trackers. All sequences come from the OTB2013 benchmark dataset: *david, car4, carDark, jogging-1, suv, girl, bolt, carScale, david3, football, skiing*.

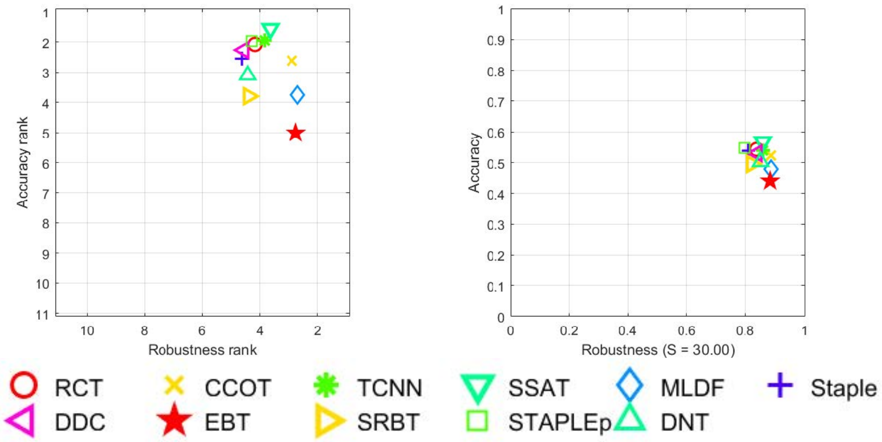


Fig. 11: A state-of-the-art comparison on the VOT2016 benchmark. In the ranking plot (left) the accuracy and robustness rank for each tracker is displayed. The AR plot (right) shows the accuracy and robustness scores.

TABLE IV: Comparison of different DCF components in RCT with Siamfc as the Siamese component.

Tracker	RCT with Staple	RCT with DSST	RCT with KCF
Mean OP (%)	74.26	65.18	60.48
Avg. FPS	46	38	34

TABLE V: Comparison of different peakiness criterion in RCT.

Peakiness Criterion	RCT with CSL	RCT with PNR
Mean OP (%)	74.26	70.12
Avg. FPS	46	55

VI. Conclusions and Future work

We present a hybrid tracker (RCT) inspired by the complementary tracking schemes of Siamfc and Staple. Siamfc and Staple are integrated into a two-stage coarse-to-fine tracking framework. An automatic activating mechanism for Siamfc is designed to achieve a real-time tracking speed. Experiments on two benchmarks demonstrate state-of-the-art performance with real-time frame-rates. We believe that considerably higher performance could be obtained by substituting the two base trackers in RCT with more advanced trackers. Moreover, the proposed tracker switch for automatically inferring tracking status from response maps of correlation filters is generic and can be incorporated into any similar tracking frameworks.

Although our method achieves significant performance improvement, it is limited to single-object tracking. Another challenge is that, if an object is completely occluded for a long period of time or if the object leaves the scene completely, our tracker will learn from incorrect samples and drift to the background. Some interesting work exploring ways to deal with these issues has been presented in [38], [39], [40] and in [41], [36], [35]. Therefore, one interesting avenue for future work would be extending our tracker to multi-object and/or long-term tracking with these inspiring ideas.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) (project no 41601487,61671456) and the Australian Research Council‘s Discovery Projects funding scheme (project no DP150104645).

References

[1] Y. Wu, J. Lim, and M. H. Yang, “Online object tracking: A benchmark,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 2411–2418. 1, 2

[2] Y. Wu, J. Lim, and M. Yang, “Object tracking benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015. 1, 2

[3] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, G. Häger, A. Lukežič, and G. Fernandez, “The visual object tracking vot2016 challenge results,” Springer, Oct 2016. 1, 2, 3, 8

[4] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol. 2, 2000, pp. 142–149 vol.2. 1

[5] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125–141, 2008. 1

[6] X. Mei and H. Ling, “Robust visual tracking using l1 minimization,” in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 1436–1443. 1

[7] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr, “Struck: Structured output tracking with kernels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, Oct 2016. 1

[8] B. Babenko, M. H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, Aug 2011. 1

[9] 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*, vol. 37, no. 3, pp. 583–596, March 2015. 1, 2, 7, 8

[10] M. Danelljan, G. Hger, F. S. Khan, and M. Felsberg, “Learning spatially regularized correlation filters for visual tracking,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4310–4318. 1, 2, 7

[11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, 2016, pp. 850–865. 1, 3, 5, 6, 7, 8

[12] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1420–1429. 1, 3, 7, 8

[13] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, 2016, pp. 749–765. 1, 3

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1. 2, 8

[15] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016. 2, 7

[16] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1090–1097. 2

[17] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Long-term correlation tracking," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5388–5396. 2, 7

[18] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, 2016, pp. 419–433. 2

[19] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1401–1409. 2, 3, 4, 6, 7, 8

[20] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2544–2550. 2

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778. 3

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016. 3

[23] X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang, "Deepsaliency: Multi-task deep neural network model for salient object detection," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3919–3930, Aug 2016. 3

[24] M. Danelljan, G. Hger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 621–629. 3, 4, 7

[25] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, 2016, pp. 472–488. 3, 7

[26] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3119–3127. 3, 4

[27] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4293–4302. 3

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. 3

[29] 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)*, Dec 2015, pp. 3074–3082. 3, 7

[30] C. Ma, Y. Xu, B. Ni, and X. Yang, "When correlation filters meet convolutional neural networks for visual tracking," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1454–1458, Oct 2016. 3

[31] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114. 3

[32] F. Porikli, "Integral histogram: a fast way to extract histogram features," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 4

[33] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692. 6

[34] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*, 2014. 7, 8

[35] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Person re-identification by unsupervised l 1 graph learning," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, 2016, pp. 178–195. 8, 10

[36] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 1239–1248. 8, 10

[37] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang, "UCT: learning unified convolutional networks for real-time visual tracking," in *2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 1973–1982. 8

[38] X. Wang, E. Türetken, F. Fleuret, and P. Fua, "Tracking interacting objects using intertwined flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2312–2326, 2016. 10

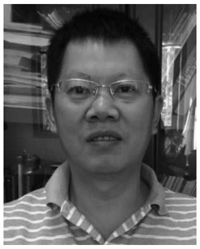
[39] A. Maksai, X. Wang, F. Fleuret, and P. Fua, "Globally consistent multi-people tracking using motion patterns," *CoRR*, vol. abs/1612.00604, 2016. 10

[40] A. Maksai, X. Wang, and P. Fua, "What players do with the ball: A physically constrained interaction modeling," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 972–981. 10

[41] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012. 10

Dongdong Li is currently pursuing his Ph.D. degree with College of Electronic Science, National University of Defense Technology (NUDT), Changsha, Hunan, China. He has been working on camera calibration, object detection and visual object tracking problems. He serves as a reviewer for Optical Engineering and Optics & Lasers in Engineering.

Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, Australia. He is also acting as the Leader of the Computer Vision Group at NICTA, Australia. He received his Ph.D. degree from NYU. Previously he served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, USA. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He is the coeditor of two books on Video Analytics for Business Intelligence and Handbook on Background Modeling and Foreground Detection for Video Surveillance. He is an Associate Editor of five journals. His publications won four Best Paper Awards and he has received the R & D 100 Award in the Scientist of the Year category in 2006. He served as the General and Program Chair of numerous IEEE conferences in the past. He has 66 granted patents..



Gongjian Wen received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1994, 1997 and 2000, respectively. Since 2009, he has been a Professor with the College of Electronic Science and Engineering, National University of Defense Technology, where he served as the head of the fourth department of the National Key Laboratory of Automatic Target Recognition. His research interests include image understanding, remote sensing and target recognition.

Yangliu Kuai received the B.S. degree and the M.S. degree from the National University of Defense Technology in 2013 and 2015 respectively. Currently, she is pursuing her Ph.D. degree with College of Electronic Science, National University of Defense Technology, Changsha, Hunan, China. SHe has been working on camera calibration, object detection and visual object tracking problems.