Collaborative Correlation Tracking

Guibo Zhu[1]
gbzhu@nlpr.ia.ac.cn
Jinqiao Wang[1]
jqwang@nlpr.ia.ac.cn
Yi Wu[2]
ywu.china@yahoo.com
Hanqing Lu[1]
luhq@nlpr.ia.ac.cn

[1] National Laboratory of Pattern Recognition,
Institute of Automation,
Chinese Academy of Sciences,
Beijing, China.

[2] B-DAT, Nanjing University of Information Science and Technology,
Nanjing, China

Abstract

Correlation filter based tracking has attracted many researchers' attention in recent years for high efficiency and robustness. Most existing works focus on exploiting different characteristics with correlation filters for visual tracking, *e.g.* circulant structure, kernel trick, effective feature representation and context information. However, how to handle the scale variation and the model drift is still an open problem. In this paper, we propose a collaborative correlation tracker to deal with the above problems. Firstly, we extend the correlation tracking filter by embedding the scale factor into the kernelized matrix to handle the scale variation. Then a novel long-term CUR filter for detection is learnt efficiently with random sampling to alleviate model drift by detecting effective object candidates in the collaborative tracker. In this way, the proposed approach could estimate the object state accurately and handle the model drift problem effectively. Extensive experiments show the superiority of the proposed method.

1 Introduction

Visual tracking is a fundamental problem in computer vision. It refers to the task of generating the trajectories of the moving objects and has many applications including surveillance, autonomous driving and image guided surgery. Numerous methods have been dedicated to generating an object trajectory by computing the translation of the object in consecutive frames, among which the correlation filter method is one of the most common methods recently [3, 6, 11, 14, 18, 38]. The popularity of the correlation filter method is due to its simplicity, high efficiency and robustness.

Correlation filter is to evaluate the similarity degree by computing the dot product for each possible alignment of one learned template (or filter) relative to a test image. After its first introduction (i.e. Person's Correlation) by Galton in 1888 [5], it has been adopted to solve various computer vision problems, such as object detection and recognition [12, 20], pose detection [13], and object tracking [3]. The computation of correlation filters can be speeded up by using the convolution theorem, which states that the convolution of two functions in the spatial domain can be computed in the Fourier domain as the element-wise multiplication of the Fourier Transform of those two functions. Due to its computational efficiency, correlation filters have attracted much attention recently for visual tracking [3, 6, 7, 14, 15, 35]. Despite its good performance, most of these correlation methods have two main limitations, the first is how to adjust the object scale efficiently. In order to consistently track the object, Danelljan *et al*. [6] proposed a separate 1-dimensional correlation filter to estimate the target scale, but they only use the original feature space as the object representation. In this paper, we propose a multi-scale kernelized correlation filter as our tracking filter by embedding the scale variation into the kernelized correlation filter while forming a separate pyramid of object representation. In addition, the use of adaptive learning rate based on failure detection is helpful for online learning a robust tracking filter.

The second limitation is how to handle the model drift problem caused by the long-term occlusion or out-of-view, which is a very important problem for online tracking [23]. One common mechanism is to introduce a detection module which can select some effective candidates to rectify the base correlation tracker. In this paper, we design a novel online CUR[1] filter for detection. CUR matrix approximation computes the low rank approximation of a given matrix by using the actual rows and columns of the matrix and have been studied in the area of theoretical computer science for large matrix approximation [8]. In the long-term tracking process, all of the historical object representations can form a large data matrix for the current frame which fits for the CUR theory. The large data matrix can be fast approximated by online CUR for representing the intrinsic object structure. In this work we develop an online CUR for learning an online detection filter by random sampling. The online CUR filter can not only exploit the low rank property of object representation [37] in the spatial-temporal domain of tracking, but also project the representation matrix of historical objects into a subspace with error upper bound so as to achieve a robust object representation. The low rank property of object representation is prevalent in long-term tracking and could be used to alleviate the model drift.

The main contributions of this work are summarized as follows:

- An efficient online CUR filter for detection is first proposed by preserving the low rank counterparts of long-term object representation, which has an error upper bound and can be computed efficiently.

- A novel collaborative correlation tracker is proposed to jointly capture the target appearance by multi-scale kernelized correlation filter and exploit the long-term object representation by the learned CUR filter.

2 Related Work

Visual tracking has been studied extensively by many researchers over the years due to its importance. While a comprehensive review of the tracking methods is beyond the scope of the paper, please refer to [22, 33] for a survey, and also to [19, 24, 26, 27, 31] for some empirical comparisons. In this section, we introduce some works closely related to this work: correlation filter based tracking and tracking-by-detection approaches.

Correlation filters have been widely studied in the field of visual tracking. Bolme *et al.* [3] modeled the target appearance by an adaptive correlation filter which was optimized by

[1] CUR approximation of a matrix A consists of three matrices, C, U, and R, where C is made from columns of A, R is made from rows of A, and that the product CUR closely approximates A.

minimizing the output sum of squared error (MOSSE). The convolution theorem can be used with correlation filters to accelerate tracking. Circulant structure with kernels tracker (CSK), proposed by Henriques *et al.* [15], exploited the circular structure of adjacent subwindows in an image for quickly learning a kernelized regularized least squares classifier of the target appearance with dense sampling. Kernelized correlation filters (KCF) [14] was an extended version of CSK by re-interpreting correlation tracking using the kernelized Ridge regression with multi-channel features. Danelljan *et al.* [7] introduced color attributes to improve tracking performance in colorful sequences and then proposed the DSST tracker [6] with accurate scale estimation by one separate filter. Zhang *et al.* [35] utilized the spatial-temporal context in the Bayesian framework to interpret correlation tracking. In a word, all of them attempt to exploit different characteristics with correlation filters for tracking, *e.g.* circular structure [15], kernel trick [14], color attributes [7], effective feature representation (*e.g.* HOG) [6, 14], the consistency of object representation in scale space [6], and context information [35].

To leverage the stability and plasticity residing online update in visual tracking, Kalal *et al.* [18] proposed a unified tracking-learning-detection (TLD) framework where short-term tracker and long-term online detector help each other by exploring the structure of unlabeled data, i.e. the short-term tracker provides high confident samples to train and update the detector, and the detector re-initializes the short-term tracker when it fails. Hare *et al.* [11] proposed structure SVM by exploring the spatial label distribution of the training samples as the intrinsic relative structure, which alleviated the problem of label prediction about noise samples (i.e. label ambiguity). Zhang and van der Maaten [36] proposed a structure preserving model with graphical structure in the tracking-by-detection framework which handled the model drift problem in some extent. Danelljan *et al.* [6] proposed a separate 1-dimensional correlation filter to estimate the target scale in an image efficiently. Henriques *et al.* [14] proposed a circular structure correlation filter tracker with kernel and interpreted the correlation tracking as a ridge regression problem which can explore the spatial label distribution with dense samples. Inspired by the above trackers, in this work we embed the scale estimation [6] into kernelized correlation filter tracker [14] as our multi-scale kernelized correlation filter tracker and propose a novel online CUR filter for detection. Due to the computational efficiency of correlation filter, the spatial label distribution by circular structure, accurate multi-scale object representations with scale estimation, and an online detection filter, the proposed tracker effectively handles the problems of label ambiguity, scale variation, and model drift existing in online tracking.

3 Collaborative Correlation Tracking

3.1 Multi-scale Kernelized Correlation Tracking (MKC)

The common idea of the correlation filter-based trackers [3, 6, 7, 14, 15, 35] is to train a discriminative correlation filter on a set of observed sample patches. The discriminative correlation filter is trained with a training sample patch X in the Fourier domain in the first frame. Then it is applied to estimate the target state in the sequential frames. After the target state is predicted, the discriminative correlation filter will be updated in each frame. Each training sample X in conjunction with a desired correlation output or probability label distribution Y in the Fourier domain is used for learning or updating the filter. The optimal kernelized correlation filter H in the Fourier domain with a fixed initialized size $\mathbf{s}_{init}$ is obtained by minimizing the following cost function,

$$\varepsilon = \|\mathcal{F}^{-1}(H \circ \varphi(X; \mathbf{s}_{init}, \mathbf{s}_{cur}) - Y)\|^2 + \lambda \|\mathcal{F}^{-1}(H)\|^2, \tag{1}$$

where $\circ$ is Hadamard product operator. $\mathbf{s}_{cur}$ is the size of the training sample in the current frame. $\varphi(X; \mathbf{s}_{init}, \mathbf{s}_{cur})$ is a mapping function for transforming the feature representation X with size $\mathbf{s}_{cur}$ into another feature representation with size $\mathbf{s}_{init}$ by preserving the consistency of multi-scale object representations in scale space. λ is a regularization parameter that controls overfitting. $\|\cdot\|$ is Frobenius norm. $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the discrete Fourier transform and the inverse discrete Fourier transform function, respectively. For simplicity, we denote $\varphi(X; \mathbf{s}_{init}, \mathbf{s}_{cur})$ as $\varphi(X)$. To estimate the object scale, multi-scale object representation similar to [6] is built independently while the predicted scale factor is embedded in Kernelized correlation filter. Therefore, the integrated tracker is denoted as multi-scale Kernelized correlation tracking.

Similar to [6], we decompose multi-scale kernelized correlation tracking into two separate filters for translation and scale estimation. Different from [6], which only used the original feature space as the object representation, we represent the object with kernel feature space and extend kernelized correlation filter with a scale factor. Based on kernel trick [25] and circular structure [15], Henriques *et al.* [14] proposed kernelized correlation filters for visual tracking which allowed more flexible, non-linear regression functions integrating with multi-channel features. Due to the characteristic of the kernel trick, the model optimization is still linear in the dual space even if with a different set of variables. Danelljan *et al.* [6] proposed a separate 1-dimensional correlation filter to estimate the target scale. Inspired by [14] and [6], we propose a multi-scale kernelized correlation filter which embeds the scale variation into the kernelized correlation filter. The multi-scale kernelized correlation tracking filter H can be represented as:

$$H = \frac{Y\Phi(\varphi(X))}{K(\varphi(X), \varphi(X)) + \lambda}, \tag{2}$$

where $\Phi(\cdot)$ is a feature mapping function to compute the kernel matrix $K(\cdot,\cdot)$ in Fourier space and X is the feature representation of the training sample in the Fourier domain.

With the guarantee of the consistency of object representation in scale space, we can scale the object representation without large loss of the intrinsic object structure. Therefore, to reduce the computational complexity and preserve the coherence of object representation in different scales, we resize the current training sample of scale $\mathbf{s}_{cur}$ to the initial scale $\mathbf{s}_{init}$ so that the feature dimension of the object filter H is consistent in the whole tracking process. The current scale $\mathbf{s}_{cur}$ is achieved independently by a feature pyramid convolution or a separate scale estimate filter similar to [6]. Therefore, our multi-scale kernelized correlation filter tracker has the characteristics of scale estimation and kernel trick, where the optimal scale $\mathbf{s}_{cur}$ can be achieved by scale estimation and multiple channel features can be embedded by kernel trick naturally.

During the tracking process, the coefficients Γ of kernelized regularized Ridge regression and the target appearance $\varphi(X)$ are updated by linear interpolation:

$$\begin{aligned} \Gamma &= \frac{Y}{K(\varphi(X), \varphi(X)) + \lambda}, & (3) \\ \Gamma^t &= (1-\beta) * \Gamma^{t-1} + \beta * \Gamma, & (4) \\ \varphi^t(X) &= (1-\beta) * \varphi^{t-1}(X) + \beta * \varphi(X), & (5) \end{aligned}$$

where t is the t-th frame and β is the learning rate. Actually, this update strategy works well when there is no occlusion and the object appearance changes slowly.

When the object is occluded, the inappropriate update of object appearance may lead to model drift . To deal with the problem, we introduce a simple indicator to evaluate whether the object is occluded and adaptively adjust the learning rate. If the object is occluded, we reduce the learning rate; if else, keep the learning rate. The indicator is the overlapping rate $\mathbb{T}_o$ between the estimated object state of multi-scale kernelized correlation tracking filter and high confident candidate bounding boxes of online detection filter. With the overlapping rate $\mathbb{T}_o$ and the lower overlapping rate bound $\mathcal{T}$, we adjust the learning rate β as follows:

$$\beta = \begin{cases} 0.1 * \beta_{init}, & if \quad \mathbb{T}_o < \mathcal{T} \\ \beta_{init}, & otherwise \end{cases} \tag{6}$$

where β_{init} is the initialization value of the learning rate β and $\mathcal{T} = 0.05$.

The new object state can be found by maximizing the correlation score s,

$$s = max\mathcal{F}^{-1}\{\Gamma \circ K(\varphi(X), \varphi(Z))\}, \tag{7}$$

where s denotes the maximum value of the confidence map of the search region $\mathbf{z} = \mathcal{F}^{-1}(Z)$ in the spatial domain, and Z is the representation of $\mathbf{z}$ in the Fourier domain.

3.2 Online CUR Filter

There is a common sense that a re-detection module is required for a robust long-term tracking algorithm in the case of tracking failure, *e.g.* out-of-view and long-term occlusion. However, how to train an effective classifier as a detector is difficult because it strongly depends on the training samples, especially for the labels of the training sample is hard to guarantee. One empirical method is to explore the spatial-temporal structure information to verify the correctness of the training sample. In addition, the time complexity of learning the classifier and using the classifier for detection with exhaustive search is high. Different from previous trackers [16, 18, 28], where online classifier needs to be trained, we propose a novel online CUR filter for detection which can be learnt easily and has very few parameters (i.e. the sampling number c) and has sufficient theory guarantee.

CUR matrix approximation is one important low rank matrix approximation technique. It computes the low rank approximation of an arbitrary data matrix by using the actual rows and columns of the matrix [29, 32]. It has been a very useful tool for handling large matrices. Specially, a CUR decomposition algorithm seeks to find a subset of c columns of A to form a matrix $C \in \mathbb{R}^{m\times c}$, a subset of r rows to form a matrix $R \in \mathbb{R}^{r\times n}$, and an intersection matrix $U \in \mathbb{R}^{c\times r}$ so that $|||A - CUR||_\xi$ is minimized. Currently, randomized algorithms (*e.g.* CUR) have been also used in the context domain of theoretical computer science and machine learning. According to the works [4, 8, 29], much tighter error bounds or much lower time complexity of the CUR algorithms are studied and guaranteed. In this paper, we propose an online CUR matrix approximation algorithm to preserve the low rank representation of the object appearance representation over time. To the best of our knowledge, it is the first work to propose an online CUR matrix approximation and apply it to object tracking to handle the model drift problem.

Before introducing the online CUR filter, we first introduce the following definition which guarantees the error upper bound of randomized projection $\mathbf{R}$.

Definition 3.1. [1] ε-isometry: Given $\varepsilon \in (0,1)$, a map $f : \mathbb{R}^p \to \mathbb{R}^q$ where $p > q$ is called an ε-isometry on set $\mathcal{X} \subset \mathbb{R}^p$ if for every pair $\mathbf{x}$, $\mathbf{y} \in \mathcal{X}$, we have

$$(1-\varepsilon)\|\mathbf{x}-\mathbf{y}\|_2^2 \leq \|f(\mathbf{x})-f(\mathbf{y})\|_2^2 \leq (1+\varepsilon)\|\mathbf{x}-\mathbf{y}\|_2^2. \tag{8}$$

We consider the case that f is denoted as a linear map $\mathbf{R} \in \mathbb{R}^{q\times p}$. The basic idea is to construct a random projection $\mathbf{R} \in \mathbb{R}^{q\times p}$ that is an exact isometry "in expectation"; that is, for every $\mathbf{x} \in \mathbf{R}^p$,

$$\mathbb{E}[\|\mathbf{Rx}\|_2^2] = \|\mathbf{x}\|_2^2. \tag{9}$$

Based on **Definition** 3.1, we just sample c columns of data matrix $\mathbf{A}$ with random sampling[2] to generate the column matrix $\mathbf{C}$, and then average the column matrix $\mathbf{C}$ so as to achieve the object detection filter $\mathbf{D}_t$. To be specific, suppose in the t-th frame we extract the object template representation matrix $\mathbf{O}_t$ inside the object bounding box which preserves the spatial corresponding relation, we vectorize the object appearance representation $\mathbf{A}_t$ to a vector $\mathbf{a}_t$ as one column of the data matrix $\mathbf{A}$, i.e. $\mathbf{A}$ can be also treated as the historical object representation matrix. After the column matrix $\mathbf{C}$ is randomly generated, we average matrix $\mathbf{C}$ in the column dimension as follows:

$$\mathbf{d}_t = \frac{1}{c}\sum_{i=1,\ldots,c}\mathbf{C}(:,i). \tag{10}$$

Then the vector $\mathbf{d}_t$ is transformed to a bounding box matrix $\mathbf{D}_t$ spatially corresponding the object template, which is treated as the CUR filter $\mathbf{D}_t$ in the current frame. After the representation $\mathbf{D}_t$ is achieved, we compute the similarity degree between $\mathbf{D}_t$ and each possible alignment in a test image using the convolution theorem. Then it can be treated as an object correlation filter. According to Eq. (11), the only parameter c depends on the value of the target rank k and the error probability ε. If we set $k = 2$ and $\varepsilon = 0.2$, we can achieve that $c \approx 20$. To preserve the intrinsic structure of data matrix $\mathbf{A}$, the randomized sampling method utilizes a common uniform sampling method and can be treated as the random projection matrix $\mathbf{R}$. **Theorem** 1 gives mathematical analysis and the theory guarantee of error upper bound for randomized selected column matrix $\mathbf{C}$ to approximate the data matrix $\mathbf{A}$. The main time complexity is the cost of generating a random number sequence, computing the average values as Eq. (10) and filtering in the test image.

Theorem 1. *[4] Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of rank ρ, a target rank $k(2 \leq k < \rho)$, and $0 < \varepsilon < 1$, the algorithm selects*

$$c = \frac{2k}{\varepsilon}(1+o(1)) \tag{11}$$

columns of $\mathbf{A}$ to form a matrix $\mathbf{C} \in \mathbb{R}^{m\times c}$. Then the following inequality holds:

$$\mathbb{E}||\mathbf{A}-\mathbf{CC}^{+}\mathbf{A}||_F^2 \leq (1+\varepsilon)||\mathbf{A}-\mathbf{A}_k||_F^2, \tag{12}$$

where the expectation is taken w.r.t. $\mathbf{C}$ and $\mathbf{C}^{+}$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{C}$, and $\mathbf{A}_k$ is the best $m \times n$ matrix of rank k constructed via the SVD. Furthermore, the matrix $\mathbf{C}$ can be obtained in time:

$$O(mk^2\varepsilon^{-\frac{r}{3}} + nk^3\varepsilon^{-\frac{2}{3}}) + T_{Multiply}(mnk\varepsilon^{-\frac{2}{3}}). \tag{13}$$

[2] To keep the results consistent from the benchmark datasets, we initialized the random number generator in matlab using the code 'stream = RandStream('mt19937ar','Seed',5489); RandStream.setGlobalStream(stream);'.

It should be noted that the detector needs to approximate the object representation data matrix $\mathbf{A}$ for the whole historical process while multi-scale kernelized correlation tracking filter pays more attention to the spatial-temporal consistency constraints between the nearest neighbour frames, i.e. the focuses of attention between a detector and a tracker are different.

3.3 Collaborative Correlation Tracker

With the multi-scale kernelized correlation tracking filter and online CUR filter, we construct a collaborative correlation tracker as follows.

In our tracking algorithm, the MKC tracker first computes the correlation output based on the previous target state. And the preliminary target state $\tilde{\mathbf{o}}_t$ (i.e. the object center location and the size of the bounding box) can be found by maximizing the correlation score. Then we detect the top-k confident bounding boxes and post-process these bounding boxes $\tilde{\mathbf{D}}_t = \{\tilde{d}_1, ..., \tilde{d}_k\}$ with the non-maximal suppression (NMS), which is a very popular post-processing method for eliminating redundant object detection windows [12]. In this paper, $k = 10$. If the overlap rate between the state $\tilde{\mathbf{o}}_t$ and one of the detected candidate bounding boxes $\tilde{\mathbf{D}}_t$ is larger than $\mathcal{T}$, we consider the state $\tilde{\mathbf{o}}_t$ as the correct target state $\mathbf{o}_t$ in the t-th frame; otherwise, the state $\tilde{\mathbf{o}}_t$ may be not correct, and then we take use of $\tilde{\mathbf{D}}_t$. To be specific, for each detection candidate bounding box we use the multi-scale kernelized correlation tracking filter to obtain the maximum correlation score $\tilde{s_i}$ and the correlation score of the preliminary target state $\tilde{s_1}$ as all candidate scores $\tilde{\mathbf{s}} = \{\tilde{s_1}, ..., \tilde{s_k}, \tilde{s}_{k+1}\}$. To preserve the spatial-temporal consistency structure in consecutive frames, we re-correct all candidate scores with spatial Gaussian distribution, which is based on the spatial distance between the candidate bounding box center and the last estimated object center. Then the corresponding candidate state of the maximum candidate correlation score is found as the final object target state $\mathbf{o}_t$. In this paper, $\mathcal{T} = 0.05$.

4 Experiments

We evaluate our collaborative tracker on two public challenging benchmark datasets, CVPR-2013 Visual Tracker Benchmark [30] and Princeton Tracking Benchmark [27], by following rigorously their evaluation protocols. There are totally 145 sequences used to evaluate the proposed approach (i.e, 50 sequences in CVPR2013 Visual Tracker Benchmark and 95 validated sequences in Princeton Tracking Benchmark). In all the experiments, we use the ***same*** parameter values for all sequences in two benchmark datasets.

We denote the proposed multi-scale kernelized correlation tracker as MKC and collaborative correlation tracker as CCT[3]. Our approaches are implemented in Matlab. The experiments are performed on an Intel(R) Core(TM) i5-2400 CPU with 2 core, 3.10 GHz and 20G RAM. In CVPR2013 Visual Tracker Benchmark, our algorithm performs well at 52.0 frames per second (FPS) average in all sequences where KCF is 175.9 FPS, DSST is 34.3 FPS, MKC is 67.9 FPS, respectively. Both our baseline MKC tracker and CCT tracker with online CUR filter are faster than DSST with better performance. Although our tracker is slower than KCF, our tracker is still real-time and our performance is better than KCF shown in Fig. 1.

[3] The source code and experimental results are available at
http://www.nlpr.ia.ac.cn/iva/homepage/jqwang/publications.htm

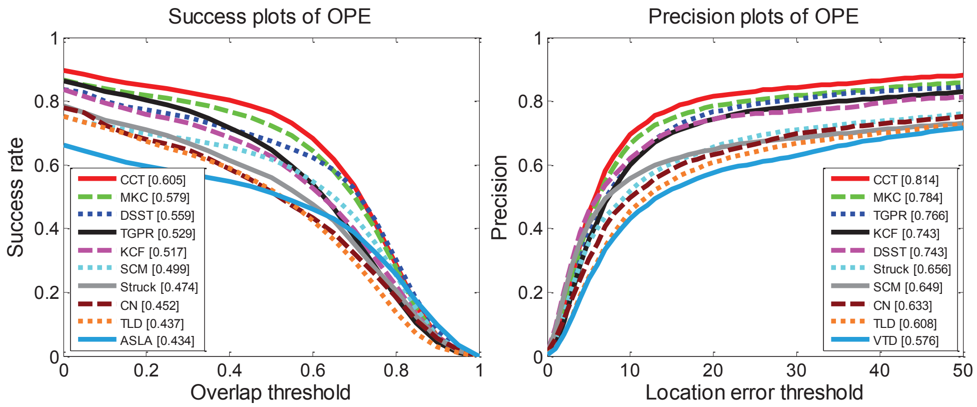

Figure 1: Precision and success plots of overall performance comparison for the 50 videos with 51 target objects in the benchmark [30] (best-viewed in high-resolution). The mean precision scores for each tracker are reported in the legends. Our methods are shown in *red* and *green*. In both cases our approaches (CCT and MKC) perform favorably better than the state-of-the-art tracking methods.

4.1 Implementation Details

To speed up the detection process, we resize the object to keep the minimum value of width or height as a small value (e.g., 32). Then we resize the test image with the same scale ratio of the object. The parameters in our multi-scale kernelized correlation tracking filter are same as [6, 14]. The CUR filter feature is represented by raw pixel values (from 0 to 1) subtracting 0.5. The sample number of column matrix $\mathbf{C}$ for learning the CUR filter is set as 20. If the row number of the data matrix $\mathbf{A}$ is smaller than 40, we update the CUR filter incrementally with the learning rate same as [6] because of simplicity.

4.2 CVPR2013 Visual Tracker Benchmark

We evaluate our methods with 33 different state-of-the-art trackers. The trackers used for comparison are: VTD [21], TLD [18], Struck [11], ASLA [17], SCM [38], CSK [15], CN [7], KCF [14], TPGR [9], DSST [6] and our trackers (MKC and CCT), etc. The overall performance is shown in Fig. 1. The public codes of the comparative trackers are provided by the authors and the parameters are fine tuning. All algorithms are compared in terms of the initial positions in the first frame from [30]. Their results are also provided with the benchmark evaluation [30] except KCF, CN, TGPR[4] and DSST. Here, KCF used HOG feature and the gaussian kernel which achieved the best performance in [14]. CN's source code was originated from [7]. It was modified to adopt the raw pixel features as [14] for handling the grey-scale images.

To evaluate the performance of the proposed method, we follow the metric used in [30], where distance precision is the relative number of frames in the sequence where the center location error of the target and the ground truth is smaller than a certain threshold (*e.g.*, 20 pixels), and overlap precision is denoted as the percentage of frames where the their

[4] The results of TGPR came from `http://www.dabi.temple.edu/~hbling/code/TGPR.htm`.

bounding box overlap exceeds a threshold (*e.g.*, 0.5). Fig. 1 shows precision and success plots which contains the mean distance and overlap precision over all the 50 sequences. The trackers in the legend are ranked using the mean precision score in precision plots and the area under the curve (AUC) in success plots, respectively. Only the top 10 trackers are displayed for clarity.

As shown in Fig. 1, our approach CCT improves the baseline HOG-based KCF tracker with a significant gain in accuracy. To be specific, our MKC and CCT tracker improves the overlap success rate of their baseline methods from 51.7% to **57.9**%, and from 57.9% to **60.5**%. Moreover, our MKC tracker improves the precision rate of the baseline method KCF from 74.3% to **78.3**% because of accurate scale estimation, and then CCT boosts the MKC tracker with a gain of **3.0**% due to online CUR filter for detection. DSST obtained the top-1 performance in the challenge of VOT2014 [19]. For merging the correlation filter tracker with kernel representation and online CUR filter for detection, our MKC and CCT tracker outperform the DSST tracker **2**% and **4.6**% in overlap success rate, and **4.1**% and **7.1**% in distance precision (20 pixels), respectively. Overall, our trackers are better than the other trackers and achieves a significant improvement.

4.3 Princeton Tracking Benchmark

Table 1: **Results on the Princeton Tracking Benchmark:** successful rates (%) and rankings (in parentheses) for different categorizations. The best results are in **red** and the second best in blue. hu.:human; an.:animal; ri.:rigid; pa.:passive;ac.:active

Algo.	**Avg.**	target type			target size		movement		occlusion		motion type	
	Rank	hu.	an.	ri.	large	small	slow	fast	yes	no	pa.	ac.
CCT	**1**	**50(1)**	**51(1)**	**64(1)**	**53(1)**	**57(1)**	**69(1)**	**50(1)**	**44(1)**	**71(1)**	**63(1)**	**53(1)**
Struck	2.82	35(2)	47(3)	53(4)	45(2)	44(4)	58(2)	39(2)	30(4)	64(2)	54(4)	41(2)
VTD	3.27	31(5)	49(2)	54(3)	39(4)	46(2)	57(3)	37(3)	28(5)	63(3)	55(3)	38(3)
RGBdet	4.36	27(7)	41(5)	55(2)	32(7)	46(3)	51(5)	36(4)	35(2)	47(6)	56(2)	34(5)
CT	5.36	31(4)	47(4)	37(7)	39(3)	34(7)	49(6)	31(5)	23(8)	54(4)	42(7)	34(4)
TLD	5.64	29(6)	35(7)	44(5)	32(6)	38(5)	52(4)	30(7)	34(3)	39(7)	50(5)	31(7)
MIL	5.82	32(3)	37(6)	38(6)	37(5)	35(6)	46(7)	31(6)	26(6)	49(5)	40(8)	34(6)
SemiB	7.73	22(8)	33(8)	33(8)	24(8)	32(8)	38(8)	24(8)	25(7)	33(8)	42(6)	23(8)
OF	9.00	18(9)	11(9)	23(9)	20(9)	17(9)	18(9)	19(9)	16(9)	22(9)	23(9)	17(9)

Princeton Tracking Benchmark was constructed by Song and Xiao [27], which consists of 100 videos with both RGB and depth data in high diverse challenging factors, including object deformation, occlusion, moving camera, and complex environments. The dataset is valuable in evaluating the effectiveness of different tracking algorithms, even if only use the RGB data.

Meanwhile, the authors also provide an online evaluation website and reserve the ground truth of 95 out of the 100 sequences for the fair comparison. Until now, there are eight state-of-the-art trackers only using RGB data and nineteen public RGBD trackers. Because we only use the RGB data, the paper compare the proposed CCT tracker with the other eight RGB trackers, including Struck [11], VTD [21], CT [34], TLD [18], MIL [2], SemiB [10], OF [27]. Table 1 shows our results generated by the website automatically after we submitted our tracking results online. The results show that the proposed CCT tracker again achieves the state-of-the-art performance over other trackers.

5 Conclusion

In this paper, we propose a collaborative correlation tracker to handle the scale variation and the model drift problem in online tracking. To be specific, multi-scale kernelized tracking filter not only better represent the object with kernel feature space, but also accurately estimate the object scale. Moreover, we develop a robust and fast CUR filter for detection which alleviates the model drift problem caused by long-term occlusion or out-of-views. Finally, extensive experiments show that our tracker outperforms the state-of-the-art methods on two tracking benchmark data sets including 145 challenging sequences.

6 Acknowledgment

This work was supported by 863 Program 2014AA015104, and National Natural Science Foundation of China 61273034, 61332016, 61370036, and 61005027.

References

[1] Rosa A., Santosh V., et al. An algorithmic theory of learning: Robust concepts and random projection. In *FOCS*, pages 616–623. IEEE, 1999.

[2] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990. IEEE, 2009.

[3] D. Bolme, J. Beveridge, B. Draper, and Y. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550. IEEE, 2010.

[4] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[5] Michael George Bulmer. *Francis Galton: pioneer of heredity and biometry*. JHU Press, 2003.

[6] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[7] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*. IEEE, 2014.

[8] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *SIAM symposium on Discrete algorithm*, pages 1117–1126. ACM, 2006.

[9] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *ECCV*, pages 188–203. Springer, 2014.

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, pages 234–247. Springer, 2008.

[11] S. Hare, A. Saffari, and P.H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, pages 263–270. IEEE, 2011.

[12] J. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *ICCV*, pages 2760–2767. IEEE, 2013.

[13] J. Henriques, P. Martins, R. Caseiro, and J. Batista. Fast training of pose detectors in the fourier domain. In *NIPS*, pages 3050–3058, 2014.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 2015.

[15] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715. Springer, 2012.

[16] Y. Hua, K. Alahari, and C. Schmid. Occlusion and motion reasoning for long-term tracking. In *ECCV*, pages 172–187. Springer, 2014.

[17] X. Jia, H. Lu, and M. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829. IEEE, 2012.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *TPAMI*, 34(7): 1409–1422, 2012.

[19] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, A. Gatt, et al. The visual object tracking vot2014 challenge results. In *ECCVW*. springer, 2014.

[20] B. V. Kumar, A. Mahalanobis, and R. Juday. *Correlation pattern recognition*. Cambridge University Press, 2005.

[21] J. Kwon and K. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276. IEEE, 2010.

[22] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A.V.D. Hengel. A survey of appearance models in visual object tracking. *TIST*, 4(4):58, 2013.

[23] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *TPAMI*, 26(6): 810–815, 2004.

[24] Y. Pang and H. Ling. Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms. In *ICCV*, pages 2784–2791. IEEE, 2013.

[25] B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[26] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2014.

[27] S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *ICCV*, pages 233–240. IEEE, 2013.

[28] JS Supancic and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, pages 2379–2386. IEEE, 2013.

[29] S. Wang and Z. Zhang. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *JMLR*, 14(1):2729–2769, 2013.

[30] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418. IEEE, 2013.

[31] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015, in press.

[32] M. Xu, R. Jin, and Z. Zhou. Cur algorithm for partially observed matrices. *arXiv preprint arXiv:1411.0860*, 2014.

[33] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *CSUR*, 38(4):13, 2006.

[34] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. In *ECCV*, pages 864–877. Springer, 2012.

[35] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *ECCV*, pages 127–141. Springer, 2014.

[36] L. Zhang and L. van der Maaten. Preserving structure in model-free tracking. *TPAMI*, 36(4):756–769, 2014.

[37] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja. Partial occlusion handling for visual tracking via robust part matching. In *CVPR*, pages 1258–1265. IEEE, 2014.

[38] W. Zhong, H. Lu, and M.H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845. IEEE, 2012.