Clustering based Ensemble Correlation Tracking

Guibo Zhu, Jinqiao Wang*, Hanqing Lu

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, 100190, China.

Abstract

Correlation filter based tracking has attracted many researchers' attention in the recent years for high efficiency and robustness. Most existing work has focused on exploiting different characteristics with correlation filter for visual tracking, *e.g.*, circulant structure, kernel trick, effective feature representation and context information. Despite much success has been demonstrated, numerous issues remain to be addressed. First, the target appearance model can not precisely represent the target in the tracking process because of the influence of scale variation. Second, online correlation tracking algorithms often encounter the model drift problem. In this paper, we propose a clustering based ensemble correlation tracker to deal with the above problems. Specifically, we extend the tracking correlation filter by embedding a scale factor into the kernelized matrix to handle the scale variation. Furthermore, a novel non-parametric sequential clustering method is proposed for efficiently mining the low rank structure of historical objects through weighted cluster centers. Moreover, to alleviate the model drift, an object spatial distribution is obtained by matching the adaptive object template learned from the clustered centers. Similar to a coarse-to-fine search strategy, the spatial distribution not only is used for providing weakly supervised information, but also is adopted to reduce the computational complexity in the detection procedure which can alleviate the model drift problem effectively. In this way, the proposed approach could estimate the object state accurately. Extensive experiments show the superiority of the proposed method.

*Corresponding author. Tel.: +8613811712597; Fax: +8601082544594.
Email address: `jqwang@nlpr.ia.ac.cn` (Jinqiao Wang)

Keywords: Object Tracking, Sequential Clustering, Correlation Filter

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 generate 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 [1, 2, 3, 4, 5, 6]. 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 convolution 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 [7], it has been adopted to solve various computer vision problems, such as object detection and recognition [8, 9], pose detection [10], correlation mining [11] and object tracking [1]. 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 filter has attracted much attention recently for visual tracking [1, 12, 13, 14, 5, 6]. Despite its good performance, most of these correlation methods have two main limitations, the first of which is how to adjust the object scale efficiently. In order to consistently track the object, Danelljan *et al.* [5] proposed a separate 1-dimensional correlation filter to estimate the target scale, but they only used 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 occlusion detection is helpful in learning a robust tracking filter online.

The second limitation is how to handle the model drift problem caused by the long-term occlusion or out-of-view [1], which is a very important problem for online tracking [15]. One feasible mechanism is to estimate the possibility of the object presence (i.e., the object spatial distribution) in a larger search region with a quick search strategy, e.g., particle filter [16]. We just need a coarse object spatial distribution. It may not be a good choice to use particle filter because it will sample many overlapping regions to evaluate the possibility with an object template or classifier which results in too many useless computation. Therefore, we choose a simple grid strategy with no-overlapping and dense samples as our auxiliary search strategy. Based on the meshing grid samples in the large search region with some evaluation metric, the object spatial distribution is computed to rectify the base correlation tracker. However, as said in [15], how to learn a good evaluation metric online is an important problem. To solve the problem, a novel non-parametric sequential clustering is proposed for efficiently mining the low rank structure of historical objects via compact cluster centers. We use an adaptive object template generated from the weighted cluster centers to represent the low rank structure and treat it as an evaluation metric. Then the spatial distribution of the object can be obtained by matching the object template and provides some weakly supervised information for re-correcting the object state. In this way, the online object tracker can exploit the low rank property of object representation [17] which is prevalent in long-term spatial-temporal tracking and is effective to alleviate the model drift.

The main contributions of this work are summarized as follows:

- A non-parametric sequential clustering is proposed for efficiently mining the low rank structure of historical objects represented by weighted cluster centers.
- To alleviate model drift, an adaptive object template is learned by the weighted clustered centers which can be used to calculate the spatial distribution of object and provide weakly supervised information for re-correcting the object state.
- A clustering based ensemble correlation tracker is proposed to jointly capture the target appearance by multi-scale kernelized correlation filter and to exploit the long-term object properties by the object template with cluster analysis.

[1] "out-of-view" means that the object of interest leaves the field-of-view and re-enters at a later time step.

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 [18, 19] for a survey, and also to [20, 21, 22, 23, 24] for some empirical comparisons. In this section, we introduce some related works closely: correlation filter based tracking, ensemble methods and tracking-by-detection approaches.

Correlation filter has been widely studied in the field of visual tracking. Bolme *et al.* [1] modeled the target appearance by an adaptive correlation filter which was optimized by minimizing the output sum of squared error (MOSSE). The convolution theorem can be used with correlation filter to accelerate tracking. Circulant structure with kernels tracker (CSK), proposed by Henriques *et al.* [12], exploited the circular structure of adjacent subwindows in an image for quickly learning a kernelized regularized least squares regressor of the target appearance with dense sampling. Kernelized correlation filters (KCF) [6] was an extended version of CSK by re-interpreting correlation tracking using the kernelized Ridge regression with multi-channel features. Danelljan *et al.* [14] introduced color attributes to improve tracking performance in colorful sequences and then proposed the DSST tracker [5] with accurate scale estimation by one separate filter. Zhang *et al.* [13] utilized the spatial-temporal context in the Bayesian framework to interpret correlation tracking. Zhu *et al.* [25] proposed a multi-scale kernel correlation tracker and online CUR filter for re-detection so as to handle the scale variation and long-term occlusion. Different from [25], this paper proposes a novel online clustering strategy for re-detection. In a word, all of them attempt to exploit different characteristics with correlation filters for tracking, *e.g.* circular structure [12], kernel trick [6], color attributes [14], effective feature representation (*e.g.* HOG) [5, 6], the consistency of object representation in scale space [5], re-detection method [25] and context information [13].

From the perspective of that the tracked objects are treated as labeled positive samples and the other as the training samples with some structure loss, the tracking problem can be considered as a supervised learning problem in each frame. Supervised learning algorithms are commonly described as performing the task of searching in a hypothesis space to find a suitable hypothesis that makes good prediction for one particular problem. Even if the hypothesis space contains many hypotheses that are very well-suited for object tracking, it may be very difficult to find a good one to locate the object precisely.

"Ensemble methods" is a machine learning paradigm where multiple (homogenous/heterogenous) individual learners are trained for the same problem, e.g., neural network ensemble [26], bootstrap aggregating (bagging) [27], boosting [28], Bayesian model averaging [29], [30], etc.. Avidan [31], who was the first to explicitly apply ensemble methods to tracking-by-detection, extended the work of [32] by adopting the Adaboost algorithm [28] to combine a set of weak classifiers maintained with an online update strategy. Along this thread, Grabner *et al.* [33] inspired from the online boosting algorithm [34] introduced feature selection from a pool of features for weak classifiers. Several other extensions to online boosting also existed, including the work by Banbenko *et al.* [35] who adopted Multiple Instance Learning in designing weak classifiers. As a different approach [36], Random Forests undergoed online update to grow or discard decision trees during tracking. Bai *et al.* [37] treated weight vector as a random variable and estimated a Dirichlet distribution for the ensemble's weight vector. They all are a binary classifier realized by an ensemble method and do not exploit the structured data properties which can improve the tracking performance significantly, like as [2], [38]. Meanwhile, online boosting based trackers [33, 35] only considered the parameter state at the current time period. Clustering methods [39, 40] are good for exploring the underlying data structure.

Zhong *et al.*[3] considered visual tracking in a weakly supervised learning scenario where (possibly noisy) labels but no ground truth are provided by multiple imperfect oracles (i.e., trackers). Kwon and Lee [41] proposed visual tracker sampler to track a target by searching for the appropriate trackers in each frame. They are all ensemble methods applied in visual tracking. Different from these methods, our method is not a heterogeneous method which focuses on the tracker space but an homogenous approach in which there is only one main tracker.

To leverage the stability and plasticity residing online update in visual tracking, Kalal *et al.* [4] 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.* [2] 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 [38] 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.* [5] proposed a separate 1-dimensional correlation filter to estimate the target scale in an image efficiently. Henriques *et al.* [6] 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 [5] into kernelized correlation filter tracker [6] as our multi-scale kernelized correlation filter tracker and propose a novel online non-parametric sequential clustering for learning an adaptive object template. 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. Clustering based Ensemble Correlation Tracking

3.1. Multi-scale Kernelized Correlation Tracking (MKC)

The basic idea of correlation filter-based trackers [1, 12, 14, 5, 6, 25] is to train a discriminative correlation filter $\mathbf{h}$ on an image patch $\{\mathbf{x}, \mathbf{y}\}$ in the first frame and then update the filter in the sequential frames, $k = \{0, ..., t-1\}$. Each image patch $\mathbf{x}$ is represented by a feature map with same spatial size $M \times N$. According to the circular structure [12], each feature map $\mathbf{x}(m, n) \in \mathbb{R}^d$ can be treated as a circular shift from $\mathbf{x}$ at each spatial location $(m, n) \in \{0, 1, ..., M-1\} \times \{0, 1, ..., N-1\}$ correspondingly.

The desired output $\mathbf{y}$ satisfies some label distribution, *e.g.* Gaussian, corresponding to all samples $\mathbf{x}(m, n)$. The correlation filter $\mathbf{h}$ in each frame can be obtained by solving the following minimization problem with l_2-loss,

$$\min_{\mathbf{h}} \sum_{m,n} \|\varphi(\mathbf{x}(m,n)) \cdot \mathbf{h} - \mathbf{y}(m,n)\|^2 + \lambda \|\mathbf{h}\|^2. \tag{1}$$

Here, φ represents the mapping function lying in a reproducing kernel Hilbert space, $\cdot$ denotes element-wise multiplication and $\lambda \geq 0$ is the impact of the regularization term. Based on Parseval's theorem, Eq. (1) in time domain can be transformed into the Fourier domain. The unitary Discrete Fourier Transform (DFT) filter $\hat{\mathbf{h}} = \mathcal{F}\{\mathbf{h}\}$ can then be solved with linear programming. According to the Representer Theorem [42], the solution $\mathbf{h}$ to the objective function can be expressed as

$$\mathbf{h} = \sum_{m,n} \alpha(m,n) \varphi(\mathbf{x}(m,n)), \tag{2}$$

where the coefficient α is defined as

$$\Gamma = \mathcal{F}\{\alpha\} = \frac{\hat{\mathbf{y}}}{\varphi(\hat{\mathbf{x}}) \cdot \varphi(\hat{\mathbf{x}})) + \lambda}, \tag{3}$$

For introducing the scale factor, we can replace $\varphi(\hat{\mathbf{x}})$ with $\varphi(\hat{\mathbf{x}}; \mathbf{s}_{init}, \mathbf{s}_{cur})$ which is a non-linear feature space (i.e., kernel trick) for transforming the feature representation $\hat{\mathbf{x}}$ 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. Here, $\mathbf{s}_{init}$ is the initialized scale size of the training sample in the first frame. $\mathbf{s}_{cur}$ is the size of the training sample in the current frame. For simplicity, we denote $\varphi(\hat{\mathbf{x}}; \mathbf{s}_{init}, \mathbf{s}_{cur})$ as $\varphi(\hat{\mathbf{x}})$. To estimate the object scale, multi-scale object representation similar to [5] 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 [5], we decompose multi-scale kernelized correlation tracking into two separate filters for translation and scale estimation. Different from [5], 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 [42] and circular structure [12], Henriques *et al.* [6] 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.* [5] proposed a separate 1-dimensional correlation filter to estimate the target scale.

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 separate scale estimate filter similar to [5]. Note that the scale estimate filter is constructed by a normalized feature pyramid with same feature dimension for fast convolution. 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 $\varphi(\hat{\mathbf{x}})$ are updated by linear interpolation:

$$\begin{aligned} \Gamma^t &= (1-\beta) * \Gamma^{t-1} + \beta * \Gamma, \quad (4) \\ \varphi^t(\hat{\mathbf{x}}) &= (1-\beta) * \varphi^{t-1}(\hat{\mathbf{x}}) + \beta * \varphi(\hat{\mathbf{x}}), \quad (5) \end{aligned}$$

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; otherwise, 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 adaptively adjust the learning rate β as follows:

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

where β_{init} is the initialization value of the learning rate β and η denotes the reducing rate for the current learning rate.

With the convolution Theorem and circulant structure [6], the correlation scores $S(\mathbf{z})$ at all locations in the image region in t-th frame can be computed efficiently,

$$S^t(z) = \mathcal{F}^{-1}\{\Gamma^{t-1} \odot (\varphi(\hat{\mathbf{z}}) \cdot \varphi(\hat{\mathbf{x}}^{t-1}))\}, \quad s_{max} = max(S(\mathbf{z})), \tag{7}$$

where the hat denotes the unitary DFT of a function and $\mathcal{F}^{-1}$ denotes the unitary inverse DFT. Then the location of s_{max} is considered as the object's state.

3.2. Online Non-parametric Sequential Clustering

In online visual object tracking, the tracked object appearance usually changes gradually. While there are some various factors such as noise or occlusion or fast and abrupt object motion or illumination changes or variations in pose and scale, the object appearance will changes much. Matthews *et al.* [15] proposed the problem that how to update the template so that it remains a good model of the tracked object. A good template update algorithm can avoid the "drifting" problem. There are many dictionary-based trackers [43, 3, 44] which maintain dictionaries of object templates and seek to represent candidate object regions in a new frame using combinations of these templates. A popular idea is using ℓ_1-norm minimization to represent the candidates sparsely. However, it is difficult to learn a good dictionary and has high time complexity. In this paper, we propose an incremental non-parametric sequential clustering method where the estimated object representation vectors are clustered into some weighted cluster centers which can be treated as the dictionary of object templates. Based on the weighted cluster centers, we can learn an adaptive object template considering with the balance between stability and plasticity. In the following, we will detail the non-parametric sequential clustering approach.

In the data stream applications (*e.g.*, online tracking), the sample vectors $X = \{\mathbf{x}_1, ..., \mathbf{x}_i, ...\}$ are presented only once. Suppose there are the learnt cluster sets $S = \{S_1, ..., S_j, ...\}$ with their corresponding cluster centers $C = \{\mathbf{c}_1, ...\mathbf{c}_j, ...\}$ and cluster sample numbers $W = \{w_1, ..., w_j, ...\}$, the number of clusters K is not known a priori. The common approach is to define the similarity function $s(\mathbf{x}_i, \mathbf{c}_j)$, set the threshold of similarity Θ and the number of maximum clusters K. Different choices for the similarity function $s(\mathbf{x}_i, \mathbf{c}_j)$ lead to different results. In this paper, the similarity function $s(\mathbf{x}_i, \mathbf{c}_j)$ is defined as follows:

$$s(\mathbf{x}_i, \mathbf{c}_j) = 0.5 * (\frac{< \mathbf{x}_i, \mathbf{c}_j >}{\|\mathbf{x}_i\| \|\mathbf{c}_j\|} + 1) \tag{8}$$

where $< \mathbf{a}, \mathbf{b} >= \mathbf{a}^T \mathbf{b}$ is the dot product of two vectors, $\|\mathbf{a}\| = \sqrt{\mathbf{a}^T \mathbf{a}}$, and $\mathbf{c}_j$ is the average of all vectors in the cluster set S_j. To reduce the time complexity, the cluster center $\mathbf{c}_j$ will be incrementally learnt as follows:

$$\mathbf{c}_j = \frac{w_j * \mathbf{c}_j}{w_j + 1} + \frac{\mathbf{x}_m}{w_j + 1} \tag{9}$$

where w_j is the number of sample in the assigned cluster $\mathbf{c}_j$ and $\mathbf{x}_m$ denotes a new sample. In our solution, the previous samples will not be stored in the memory while just keep the cluster centers C and cluster sample numbers W. The cluster sample numbers W can be transformed into the weights of the cluster centers.

To be specific, the idea is to assign each newly sample vector to an existing cluster or create a new cluster for this sample, depending on the similarity to the already learnt clusters. In the context of online tracking, the sample vector changes gradually so that the threshold Θ and the number K are difficult to set. Herein, to avoid the above initialized settings, we do not set the upper bound of the number K so that the clustering algorithm becomes a non-parametric one. If the maximum similarity between the sample and the cluster centers is lower than the threshold Θ, a new cluster will be created; otherwise, the sample will be assigned to one nearest cluster center. As shown in Algo. 1, it is implemented as follows:

In each frame, non-parametric sequential clustering will generate the weighted cluster centers $C = \{\mathbf{c}_1, ..., \mathbf{c}_m\}$ with their corresponding weights $W = \{w_1, ..., w_m\}$. Then we use the weighted cluster centers to learn an adaptive object template. To make

Algorithm 1 Non-parametric Sequential Clustering

```
Initialize the first sample as the first cluster set $S_1 = \{\mathbf{x}_1\}$, the cluster center $\mathbf{c}_1 = \mathbf{x}_1$, the cluster sample number $w_1 = 1$, the threshold of the similarity $\Theta = 0.85$;
for each sample vector $\mathbf{x}_i \in \{\mathbf{x}_2, ..., \mathbf{x}_n, ...\}$ in the data stream do
    Find the most similar cluster $S_k$ according to Eq. (8) so that the similarity $\alpha = \max s(\mathbf{x}_i, \mathbf{c}_j)$;
    if $\alpha < \Theta$ then
        Create a new cluster $S_m = \{\mathbf{x}_i\}, \mathbf{c}_m = \mathbf{x}_i, w_m = 1$;
    else
        Add the sample $\mathbf{x}_i$ to the nearest cluster $S_k = \{S_k, \mathbf{x}_i\}$, the cluster sample number $w_k = w_k + 1$, and compute the cluster center $\mathbf{c}_k$ according to Eq. (9).
    end if
end for
```

the object template robust to the outliers caused by occlusion or out-of-view, we first sort the weights W in descending order to get the median weight w_c and then select the cluster centers whose weights are no less than w_c to generate the adaptive object template $\mathbf{o}$ as follows.

$$\mathbf{o} = \frac{\sum_{i=1,\ldots,m} \delta(w_i - w_c) w_c \mathbf{c}_i}{\sum_{i=1,\ldots,m} \delta(w_i - w_c) w_c}, \tag{10}$$

$$\delta(x) = \begin{cases} 1, & if \quad x >= 0 \\ 0, & if \quad x < 0 \end{cases} \tag{11}$$

3.3. Clustering based Detector

There is a common sense that a re-detection module is necessary for a robust long-term tracker in the case of tracking failure, *e.g.* out-of-view and long-term occlusion. However, how to train an effective detector is difficult because it strongly depends on the training samples, especially the labels of the training samples are hard to guarantee. An 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 is high, and using the classifier for detection with exhaustive search is also time-consuming. Different from previous trackers [4, 45, 46], where the online classifier needs to be trained, we propose a online non-parametric sequential clustering for learning an object template adaptively as a detector only with one parameter (i.e., the threshold of similarity Θ to create new clusters). Based on Sec. (3.2), we learn the clustered object template $\mathbf{o}$ and treat it as clustering based detector.

The clustering based detector is used to calculate a spatial distribution of object location. Firstly, we extract the sample feature vectors $Z = \{\mathbf{z}_1, ..., \mathbf{z}_P\}$ centred around the previous object location center using a grid strategy where the size of each sample is the same as the previous estimated object scale $\mathbf{s}_{cur}$. The grid number P determines the context region or the larger search region. In the paper, the grid is 9×9. Then we compute the similarities $\mathbf{s} = \{s_1, ..., s_P\}$ between the extracted vectors Z and the object template $\mathbf{o}$ in Eq. (10) according to Eq. (8). We treat the similarities $\mathbf{s}$ as the spatial distribution of object learned from clustering based detector.

It should be noted that the clustering based detector is to approximate object representation vectors in 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.

3.4. Clustering based Ensemble Correlation Tracker

With the multi-scale kernelized correlation tracking filter and the spatial distribution of object, we construct a clustering correlation tracker as follows.

In our tracking algorithm, the MKC tracker first computes the correlation output based on the previous target state. Then the preliminary target state $\tilde{\mathbf{o}}_t$ (i.e., the object center location and the size of the bounding box) is achieved by maximum response estimation. Based on the similarities $\mathbf{s}$ in Sec. 3.3, we only keep the top-k samples whose similarities exceed over 0.5. The kept samples will provides new candidate centers. Based on the location centers of the top-k samples, we get some bounding boxes $\tilde{\mathbf{D}}_t = \{\tilde{d}_1, ..., \tilde{d}_k\}$. In this paper, $k = 3$. If the overlap rate between the state $\tilde{\mathbf{o}}_t$ and one of the bounding boxes $\tilde{\mathbf{D}}_t$ is larger than $\mathcal{T}$, and the similarity between $\tilde{\mathbf{o}}_t$ and the clustered template $\mathbf{o}$ is the largest, we consider the state $\tilde{\mathbf{o}}_t$ as the correct target state $\mathbf{o}_t$ in the t-th frame; otherwise, the preliminary target state $\tilde{\mathbf{o}}_t$ may be not correct, and then we take use of $\tilde{\mathbf{D}}_t$. To be specific, for each candidate bounding box we use the multi-scale kernelized correlation tracking filter to obtain the correlation score $\tilde{s_i}$. Then the scores of the preliminary state $\tilde{s_1}$ and the top-k candidate scores constructed the total 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 the object spatial distribution and spatial Gaussian distribution. The spatial Gaussian distribution 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 score is chose as the final object target state $\mathbf{o}_t$.

4. Experiments

We evaluate our collaborative tracker on two public challenging benchmark data sets, Online Tracking Benchmark (OTB) [21] and Princeton Tracking Benchmark [22], by following their evaluation protocols rigorously. There are totally 145 sequences used to evaluate the proposed approach (i.e., 50 sequences in OTB and 95 validated sequences in Princeton Tracking Benchmark). There are seven sequences with more than 1000 frames and 19 sequences with more than 500 frames in OTB. 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 clustering based ensemble correlation tracker as CECT. 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 OTB, our algorithm performs well at 12.0 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.

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

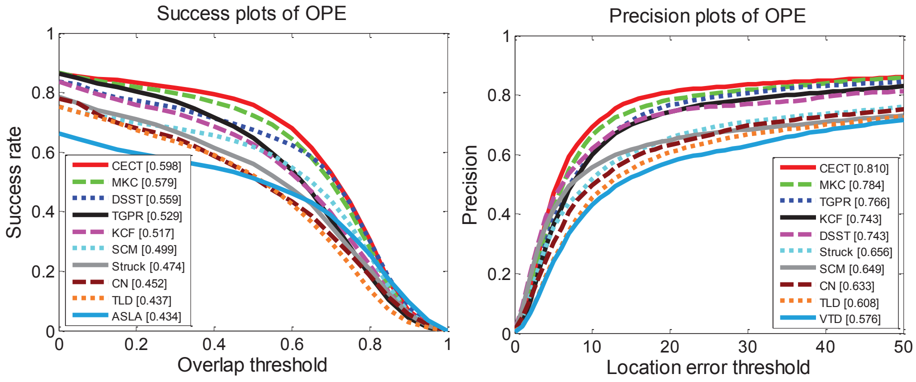

Figure 1: Precision and success plots of overall performance comparison for the 50 videos with 51 target objects in the benchmark [21] (best-viewed in high-resolution). The mean precision scores for each tracker are reported in the legends. Our methods are shown in *red* and *green*. In both cases our approaches (CECT and MKC) perform favorably better than the state-of-the-art tracking methods. OPE denotes one-pass evaluation [21].

same scale ratio of the object. The parameters in our multi-scale kernelized correlation tracking filter are same as [6, 5]. The object template feature is represented by color histograms. To consider the local color distribution of the target object, its bounding box is divided into 2×2 blocks, and the CIE Lab color histogram with 48 bins is extracted from each block. Hence the feature vector has the dimension of 192. If the sequence is gray-scale, the feature vector has the dimension of 64. In Eq. (6), $\mathcal{T} = 0.05$ and $\eta = 0.1$, which are mainly set empirically without too much research.

4.2. OTB

We evaluate our methods with nine state-of-the-art trackers. The trackers used for comparison are: VTD [41], TLD [4], Struck [2], ASLA [47], SCM [3], CN [14], KCF [6], TPGR [48], DSST [5] and our trackers (MKC [25] and CECT). 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 coming from [21]. Their results are also provided with the benchmark evaluation [21] except KCF, CN, TGPR[2] and DSST. Here, KCF used HOG feature and the gaussian kernel which achieved the best performance in [6]. CN's source code was originated from [14]. It was modified to adopt the raw pixel features as [6] for handling the grey-scale images.

To evaluate the performance of the proposed method, we follow the metric used in [21], 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 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.

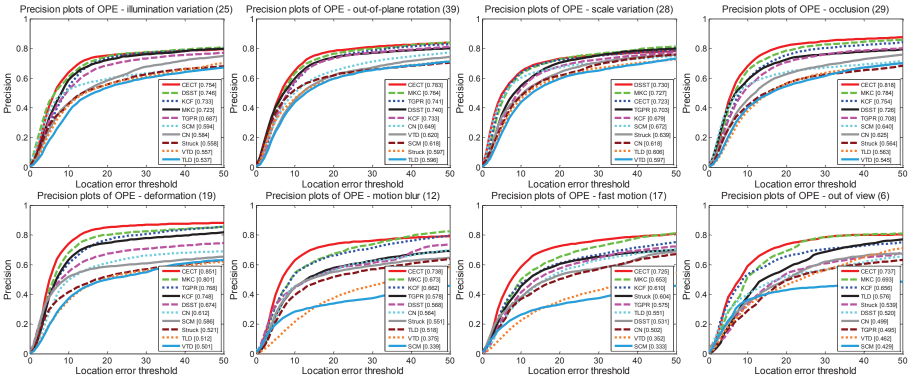

Figure 2: Precision plots of different attributes (best-viewed on high-resolution display) generated by the toolkit [21]. The valued appearing in the title denotes the number of videos associated with the respective attribute. The proposed methods in this paper perform favorably against state-of-the-art algorithms.

As shown in Fig. 1, our approach CECT improves the baseline HOG-based KCF tracker with a significant gain in accuracy. To be specific, our MKC and CECT tracker improves the overlap success rate of their baseline methods from 51.7% to $\mathbf{57.9\%}$, and from 57.9% to $\mathbf{59.8\%}$. Moreover, our MKC tracker improves the precision rate of the

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

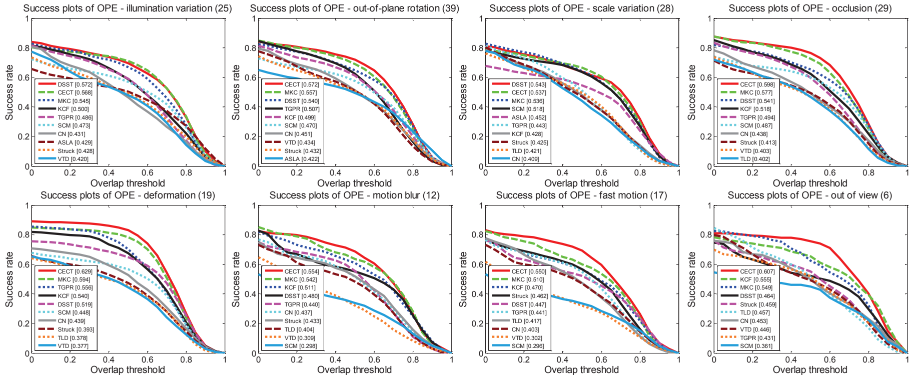

Figure 3: Success plots of different attributes generated by the toolkit [21]. The proposed methods (MKC and CECT) obtain better or comparable performance in all the subsets.

baseline method KCF from 74.3% to $\mathbf{78.4}\%$ because of accurate scale estimation, and then CECT boosts the MKC tracker with a gain of $\mathbf{2.6}\%$ due to the object template learnt by cluster analysis. DSST, which has shown to obtain the top-1 performance in the challenge of VOT2014 [24] than most of the state-of-the-art trackers. For merging the correlation filter tracker with kernel representation and an adaptive object template with clustering for detection, our MKC and CECT tracker outperform the DSST tracker $\mathbf{2}\%$ and $\mathbf{3.9}\%$ in overlap success rate, and $\mathbf{4.1}\%$ and $\mathbf{6.7}\%$ in distance precision (20 pixels), respectively. Overall, our trackers are better than the other trackers and achieves a significant improvement. Certainly, according to the attribute of scale variation in Fig. 2 and Fig. 3, we find that DSST is better than the proposed trackers MKC and CECT in handling the problem of scale variation.

Attribute-based Evaluation: There are several factors which can affect the performance of a visual tracker. In the recent benchmark evaluation [21], the sequences are annotated with 11 different attributes, which are named as: occlusion, deformation, illumination variation, fast motion, motion blur, out-of-plane rotation, scale variation, background clutter, out-of-view, low resolution and in-plane rotation. These sequence subsets with different dominant attributes can facilitate the analysis of the performance of trackers for each challenging factor. Fig. 2 and Fig. 3 show example precision plots and success plots of different attributes. Different methods have their special charac-

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.

Algo.	**Avg.**	target type			target size		movement		occlusion		motion type	
	Rank	human	animal	rigid	large	small	slow	fast	yes	no	passive	active
CECT	**1.09**	**58(1)**	53(2)	**65(1)**	**61(1)**	**58(1)**	**71(1)**	**55(1)**	**51(1)**	**71(1)**	**66(1)**	**57(1)**
KCF	1.91	40(2)	**55(1)**	63(2)	45(2)	56(2)	63(2)	47(2)	39(2)	68(2)	63(2)	47(2)
Struck	3.82	35(3)	47(4)	53(5)	45(3)	44(5)	58(3)	39(3)	30(5)	64(3)	54(5)	41(3)
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)

teristics in different attributes. Please refer to [21] for more details.

As shown in Fig. 2, CECT provides superior results compared to existing methods in the following attributes, including illumination variation, out-of-plane rotation, motion blur, occlusion, deformation and so on, mainly because of the grid strategy with the adaptive object template by cluster analysis. The object template extends the search region with some guarantee by non-parametric sequential clustering. The comparison of the object template and the search grid strategy can re-correct some object states.

4.3. Princeton Tracking Benchmark

Princeton Tracking Benchmark was constructed by Song and Xiao [22], which consisted 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 CECT tracker with the other eight RGB trackers, including Struck [2], VTD [41], CT [49], TLD [4], MIL [35], SemiB [50], OF [22]. Table 1 shows our results generated by the website automatically after we submitted our tracking results online. The results show that the proposed CECT tracker again achieves the state-of-the-art performance over other trackers.

5. Conclusion

In this paper, we propose a clustering based ensemble correlation tracker to handle the scale variation and model drift 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 non-parametric sequential clustering for learning an adaptive object template which extends the search region and alleviates the model drift caused by 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, and 61332016.

References

[1] D. Bolme, J. Beveridge, B. Draper, Y. Lui, Visual object tracking using adaptive correlation filters, in: CVPR, IEEE, 2010, pp. 2544–2550.

[2] S. Hare, A. Saffari, P. Torr, Struck: Structured output tracking with kernels, in: ICCV, IEEE, 2011, pp. 263–270.

[3] W. Zhong, H. Lu, M. Yang, Robust object tracking via sparsity-based collaborative model, in: CVPR, IEEE, 2012, pp. 1838–1845.

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

[5] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: BMVC, 2014.

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

[7] M. G. Bulmer, Francis Galton: pioneer of heredity and biometry, JHU Press, 2003.

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

[9] J. Henriques, J. Carreira, R. Caseiro, J. Batista, Beyond hard negative mining: Efficient detector learning via block-circulant decomposition, in: ICCV, IEEE, 2013, pp. 2760–2767.

[10] J. Henriques, P. Martins, R. Caseiro, J. Batista, Fast training of pose detectors in the fourier domain, in: NIPS, 2014, pp. 3050–3058.

[11] Z. Li, J. Liu, C. Xu, H. Lu, Mlrank: Multi-correlation learning to rank for image annotation, PR 46 (10) (2013) 2700–2710.

[12] J. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: ECCV, Springer, 2012, pp. 702–715.

[13] K. Zhang, L. Zhang, Q. Liu, D. Zhang, M. Yang, Fast visual tracking via dense spatio-temporal context learning, in: ECCV, Springer, 2014, pp. 127–141.

[14] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, Adaptive color attributes for real-time visual tracking, in: CVPR, IEEE, 2014.

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

[16] M. Isard, A. Blake, Condensationconditional density propagation for visual tracking, IJCV 29 (1) (1998) 5–28.

[17] T. Zhang, K. Jia, C. Xu, Y. Ma, N. Ahuja, Partial occlusion handling for visual tracking via robust part matching, in: CVPR, IEEE, 2014, pp. 1258–1265.

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

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

[20] Y. Pang, H. Ling, Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms, in: ICCV, IEEE, 2013, pp. 2784–2791.

[21] Y. Wu, J. Lim, M. H. Yang, Online object tracking: A benchmark, in: CVPR, IEEE, 2013, pp. 2411–2418.

[22] S. Song, J. Xiao, Tracking revisited using rgbd camera: Unified benchmark and baselines, in: ICCV, IEEE, 2013, pp. 233–240.

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

[24] 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.

[25] G. Zhu, J. Wang, Y. Wu, H. Lu, Collaborative correlation tracking, in: BMVC, 2015.

[26] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE-TPAMI 12 (10) (1990) 993–1001.

[27] L. Breiman, Bagging predictors, Machine Learning 24 (1997) 123–140.

[28] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Jouranl of Computer and System Sciences 55 (1) (1997) 119–139.

[29] A. Jennifer, M. David, E. Adrian, T. Chris, Bayesian model averaging: A tutorial, Statistical Science 14 (4) (1999) 382–417.

[30] S. Hong, S. Kwak, B. Han, Orderless tracking through model-averaged posterior estimation, in: ICCV, IEEE, 2013.

[31] S. Avidan, Ensemble tracking, IEEE-TPAMI 29 (2) (2007) 261–271.

[32] S. Avidan, Support vector tracking, IEEE-TPAMI 26 (8) (2004) 1064–1072.

[33] H. Grabner, H. Bischof, On-line boosting and vision, in: CVPR, Vol. 1, IEEE, 2006, pp. 260–267.

[34] N. C. Oza, Online bagging and boosting, in: Systems, man and cybernetics, 2005 IEEE international conference on, Vol. 3, IEEE, 2005, pp. 2340–2345.

[35] B. Babenko, M. H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: CVPR, IEEE, 2009, pp. 983–990.

[36] A. Saffari, C. Leistner, J. Santner, M. Godec, H. Bischof, On-line random forests, in: ICCVW, IEEE, 2009, pp. 1393–1400.

[37] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, C. Monnier, Randomized ensemble tracking, in: ICCV, IEEE, 2013.

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

[39] Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data representation, TPAMI 37 (13) (2015) 2085 – 2098.

[40] Z. Li, J. Liu, Y. Yang, X. Zhou, H. Lu, Clustering-guided sparse structural learning for unsupervised feature selection, TKDE 26 (9) (2014) 2138–2150.

[41] J. Kwon, K. Lee, Visual tracking decomposition, in: CVPR, IEEE, 2010, pp. 1269–1276.

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

[43] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, Minimum error bounded efficient l1 tracker with occlusion detection, in: CVPR, IEEE, 2011, pp. 1257–1264.

[44] N. Wang, J. Wang, D. Yeung, Online robust non-negative dictionary learning for visual tracking, in: ICCV, IEEE, 2013, pp. 657–664.

[45] J. Supancic, D. Ramanan, Self-paced learning for long-term tracking, in: CVPR, IEEE, 2013, pp. 2379–2386.

[46] Y. Hua, K. Alahari, C. Schmid, Occlusion and motion reasoning for long-term tracking, in: ECCV, Springer, 2014, pp. 172–187.

[47] X. Jia, H. Lu, M. Yang, Visual tracking via adaptive structural local sparse appearance model, in: CVPR, IEEE, 2012, pp. 1822–1829.

[48] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with gaussian processes regression, in: ECCV, Springer, 2014, pp. 188–203.

[49] K. Zhang, L. Zhang, M. Yang, Real-time compressive tracking, in: ECCV, Springer, 2012, pp. 864–877.

[50] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: ECCV, Springer, 2008, pp. 234–247.