Robust visual tracking by exploiting the historical tracker snapshots

Jiatong Li[1,2], Zhibin Hong[2], and Baojun Zhao[1]

[1]School of Information and Electronics, Beijing Institute of Technology
5 South Zhongguancun Street, Haidian District, Beijing
[2]Faculty of Engineering and Information Technology, University of Technology, Sydney
81 Broadway, Ultimo NSW 2007, Australia
{Jiatong.Li-3@student., Zhibin.Hong@student.}uts.edu.au, zbj@bit.edu.cn

Abstract

Variations of target appearances due to illumination changes, heavy occlusions and abrupt motions are the major factors for tracking failures. In this paper, we show that these failures can be effectively handled by exploiting the trajectory consistency between the current tracker and its historical trained snapshots. Here, we propose a Scale-adaptive Multi-Expert (SME) tracker, which combines the current tracker and its historical trained snapshots to construct a multi-expert ensemble. The best expert in the ensemble is then selected according to the accumulated trajectory consistency criteria. The base tracker estimates the translation accurately with regression based correlation filter, and an effective scale adaptive scheme is introduced to handle scale changes on-the-fly. SME is extensively evaluated on the 51 sequences tracking benchmark and VOT2015 dataset. The experimental results demonstrate the excellent performance of the proposed approach against state-of-the-art methods with real-time speed.

1. Introduction

Visual tracking is one of the fundamental problems among numerous research topics in computer vision. A common tracking scenario is to track the unknown object given only the initial bounding box of the target. This problem is a challenging task due to target deformations, illumination variations, abrupt motions, partial occlusions and background clutters.

To handle tracking failures caused by the above mentioned factors, a commonly used strategy is to design an online model that evolves forward to adapt to the target appearance changes. The main drawback of this method is that online models tend to drift with the time passing by. The drift happens even more easily due to large appearance changes of the object, abrupt motions and heavy occlusions. To tackle the model drift problem, many methods propose to use tracker ensembles which are composed of more than one base trackers to determine the target position [17, 18, 25, 15, 24]. One strategy is to establish a tracker pool and choose the most appropriate tracker each frame to make the best decision [17, 18, 25]. Others use multiple experts working parallel to better discriminate the target from the background [15, 24]. One of the representative work is that in [29], Zhang *et al*. propose to use the multi-expert restoration scheme to address the model drift problem, where an entropy based loss function is defined to

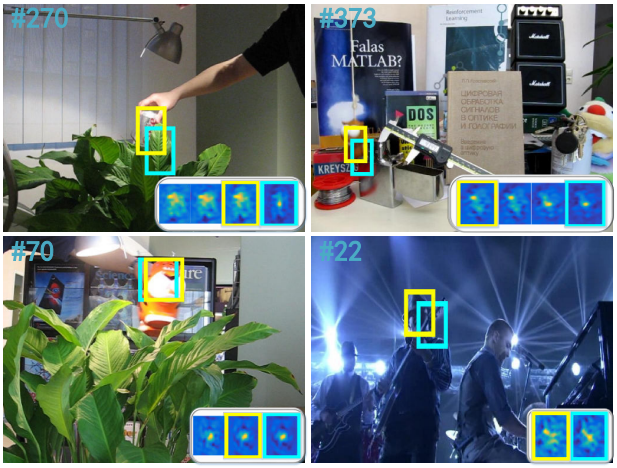

Figure 1: Four typical sequences (*Coke*, *Lemming*, *Tiger1*, *Shaking*) show the importance of exploiting the historical tracker snapshots. The cyan bounding box is the tracking result of our base tracker, while the yellow box is the selected tracker historical snapshot by the multi-expert framework. The tracker snapshots are stored at the pre-defined interval, and the corresponding response maps are illustrated at the bottom-right of the image (listed in chronological order from left to right). The color brightness of the response map indicates the confidence degree of the tracker snapshot. The number at the top-left corner is the frame count.

determine whether the current tracker is reliable and should be restored to the historical tracker. The proposed tracking framework adopts online SVM as the base tracker, which shows very robust performance. The work by Zhang indicates that, to some extent, the historical trained trackers, also called tracker snapshots, can be used to prevent the model drift.

The key motivation of our method is the observation that tracking failures can be effectively handled by exploiting the historical tracker snapshots. As shown in Fig. 1, during the tracking process, the object goes through significant appearance variation, occlusion and illumination change. Therefore, the current tracker tends to drift (the response map is framed by the cyan box). However, it is observed that the target location can be accurately estimated by most of the historical tracker snapshots. For example, in sequence *Coke*, after the object having been occluded by the leaves, the current tracker is distracted by the background, but its three past snapshots are all able to identify the true target. The same phenomenon is shown in sequence *Lemming* and *Tiger1*. In addition, as illustrated in sequence *Shaking*, we will show that by designing the appropriate snapshot selection criteria, the early period tracking failure can also be avoid.

The above phenomenon gives the main insight of our work. During the tracking process, single tracker is easy to overfit when there is target partial occlusion, abrupt motion, background clutter and other factors lead to object appearance variation. However, the above moments are relatively short during the whole tracking process. On the other hand, the diversity of target appearance is usually limited, and cannot be varying significantly all the time without restore to is past appearance. Therefore, sometimes the past tracker snapshot is capable to recognize the object better than the current tracker. Particularly, the past snapshot can re-identify the target after its occlusion and abrupt motion, which is naturally to rescue the tracking failure. As a result, in this paper, we exploit the historical tracker snapshots and show that tracking performance can be effectively promoted by exploiting the relationship between them.

The main contribution is that we propose a trajectory consistency based Scale-adaptive Multi-Expert (SME) tracking framework. The multi-expert ensemble is constituted by the current tracker and its past trained snapshots. Moreover, we adopt the regression model based correlation filter as the base tracker, which is used to learn the temporal context correlation of the target. Benefit from the Discrete Fourier Transform, correlation filter learns all the circular shift of the extended image patches containing the target, while maintains low computation load. On the widely used 51 sequences benchmark [27], SME gives significant improvement against other state-of-the-art methods. The proposed tracker is further tested on the new VOT2015 dataset, which also shows its excellent performance.

2. Related Work

Visual tracking has been extensively studied [20, 28]. Recent public available benchmark and evaluation have also accelerated the development of this field [27, 23, 16].

The tracking-by-detection method plays a key role among numerous recent tracking methods. Under this framework, a discriminative classifier is trained to classify the foreground and background features [1, 2, 3, 30, 10]. For instance, Avidan [1] integrates the SVM classifier into the optical flow to establish the online target discriminative model. Babenko *et al*. [3] introduce multiple instance learning to collect positive and negative samples into bags to avoid model drift. In [30], random projection is used to reduce feature dimension which achieves real-time tracking. Particularly, Hare *et al*. [10] use structured output SVM and trains samples with structured labels, which shows excellent performance in the benchmark [27].

Recently, correlation filter based tracking methods have attracted great attention due to its high efficiency [4, 11, 12, 7, 6, 22]. Since Bolme *et al*. [4] propose a minimum output sum of squared error filter for tracking, correlation filter began to re-attract attention as a commonly used method in signal processing. After that, Henriques *et al*. [11] propose to use circular image patches as dense samples to train the correlation filter in kernel space with low computation load. The above methods are based on gray-level feature. The work is further extended to HOG feature in the KCF tracking algorithm [12]. In [7], color attributes are added to the framework of [11], and an adaptive dimension reduction technique is proposed, which demonstrates the importance of color feature in visual tracking. Other extended work, such as in [6, 21, 13, 22], the scale variation, long term tracking, even long short term memory scheme are added to the correlation filter tracker. In [6], the accurate scale estimation is obtained by treating translation and scale correlation separately, and a one-dimensional scale correlation filter is used to measure scale change.

Some tracking algorithms adopt tracker ensemble to achieve more robust tracking performance. For example, Kwon [17] decomposes traditional Bayesian recursive framework into basic models, and uses the MCMC sampling to integrate them. In [18], the proposed method not only samples the target state space but also the tracker space to handle challenging tracking scenes. Kalal *et al*. [15] address the long term tracking problem by designing two complementary experts, one estimates missed detections and the other estimates false alarm, apart from this, a re-detection scheme is designed to achieve long term tracking. In [25], a sparsity-based collaboration of discriminative and generative modules are proposed. Hong *et al*. [14] adopt the hierarchical appearance model to track object through multi-

level. In MEEM tracker [29], multiple experts are used to handle the model drift problem, which shows high tracking efficiency.

Our work is most close to MEEM [29], but with significant differences summarized as follows. Firstly, in [29], the online SVM method is adopted as the base tracker, and the grid searching method is used to sample image patches. Our method introduces the ridge regression model to learn the temporal context correlation of the object rather than the binary classifier (online SVM). Secondly, in [29], multiple experts are regarded independently, and the entropy based loss function is defined on the single expert. Furthermore, since our base tracker of correlation filter uses the regression model with dense sampling scheme, the response map shows much less ambiguity than the binary classifier. Therefore, only the entropy based loss function will not work in our method. As a result, in this paper, we further pay more attention to the collaborative efforts of multi-expert rather than the single one, and propose the trajectory consistency based multi-expert selection criteria. Finally, we additionally take target scale variation into consideration which [29] cannot deal with.

3. The proposed tracker

In this section, we first introduce the multi-expert ensemble framework and the expert selection criteria, and then presents the base tracker of scale adaptive correlation filter.

3.1. Multi-Expert Selection

Given a base tracker which updates every frame, let $\mathcal{T}_t$ denotes the tracker snapshot (expert) trained up to time t (In the following, we do not differentiate tracker snapshot from expert). Until time T, we have an expert ensemble $\mathbf{E} := \{\mathcal{T}_{t_1}, \mathcal{T}_{t_2}, ..., \mathcal{T}_T\}$, where $\mathcal{T}_T$ represents the tracker at the current time. At each time step, a score is calculated and assigned to each expert in the ensemble, the best expert is determined by its accumulative score within a pre-defined temporal window:

$$\mathcal{T}^* = arg\max_{\mathcal{T}\in\mathbf{E}} \sum_{t\in[T-\Delta,T]} \mathcal{S}^t_{\mathcal{T}}, \tag{1}$$

where $\mathcal{S}^t_{\mathcal{T}}$ is the score of expert $\mathcal{T}$ at time t, and Δ is the temporal window size.

It is very important to define the expert score. Trajectory analysis is an effective method used in tracking. Kalal [15] proposes to use the forward-backward trajectory consistency of the optical flow to determine the robustness of the tracker. In [19], multiple trajectories obtained by different features are used and the best tracker is selected by their reliability, which achieves very good performance. Inspired by these work, we define the trajectory consistency score of the expert. Let $\overrightarrow{x}_{\mathcal{T}}(t)$ denotes the position of the bounding box center determined by expert $\mathcal{T}$ at time step t, then the trajectory from time t_1 to t_2 is be denoted by:

$$\overrightarrow{X}_{\mathcal{T}}(t_1 : t_2) = \{\overrightarrow{x}_{\mathcal{T}}(t_1), \overrightarrow{x}_{\mathcal{T}}(t_1 + 1), ..., \overrightarrow{x}_{\mathcal{T}}(t_2)\}. \tag{2}$$

We measure the consistency of trajectories according to their position similarity. Given two trajectories determined by expert $\mathcal{T}_1$ and expert $\mathcal{T}_2$, their position similarity at time t is defined as:

$$\mathcal{C}^t_{\mathcal{T}_1:\mathcal{T}_2} = \exp(-\frac{\|\overrightarrow{x}_{\mathcal{T}_1}(t) - \overrightarrow{x}_{\mathcal{T}_2}(t)\|^2}{\sigma^2}). \tag{3}$$

Assuming there are n experts, then the trajectory consistency score of expert $\mathcal{T}$ at time t is the mean of its position similarity relative to all the other experts:

$$C^t_{\mathcal{T}} = \frac{1}{n-1} \sum_{(\mathcal{T}_i\in\mathbf{E})\bigcap(\mathcal{T}_i\neq\mathcal{T})} \mathcal{C}^t_{\mathcal{T}:\mathcal{T}_i}\ . \tag{4}$$

The above score definition favors the expert whose trajectory is more consistent with the other experts. In actual tracking scenario, the expert tends to be ambiguous due to successive target appearance variations, especially when there is background clutter, heavy occlusion and abrupt motion. To further enhance the expert selection criteria, we add the entropy based regularization term in the score as the prior [9, 29], so as to give penalty to the tracker ambiguity. For convenience, we omit the superscript of time t in the following equations. Taken the entropy prior into consideration, the whole score represents the log posterior of the expert is denoted by:

$$\mathcal{S}_{\mathcal{T}} = L_{\mathcal{T}} - \eta H_{\mathcal{T}}(Y|X, Z), \tag{5}$$

where $L_{\mathcal{T}}$ is the natural logarithm of the trajectory consistency score denoted in Eq. 4, which can be regarded as the log likelihood, the scalar η is the regularization coefficient to control the tradeoff between the two terms, and the entropy regularization is computed by:

$$H_{\mathcal{T}}(Y|X, Z) = -\sum_{Y\in Z} P(Y|X; \theta_{\mathcal{T}}) \log P_{\mathcal{T}}(Y|X; \theta_{\mathcal{T}}). \tag{6}$$

In the above equation, X is the possible target candidates proposed by the expert ensemble $\mathbf{E}$, and Z represents the possible label set containing the true label Y of X.

To be more specific, at each time step, the expert ensemble $\mathbf{E}$ proposes a target candidates set $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, each $\mathbf{x}_i$ is an image patch sampled within the search window, which is labelled by l_i, where $l_i \in \{1, -1\}$ denotes the foreground and background label. The candidates are sparsely distributed and do not heavily overlap with each other, so it is assumed one of the samples in X is the true target. Therefore, the ground truth Y is included in a small possible

label set $Z = \{Y^1, Y^2, ..., Y^n\}$, where $Y^j = (l_1^j, l_2^j, ..., l_n^j)$, and l_i^j is the positive label only when $i = j$, meaning that only one sample in X is the true target. Since the target candidates do not substantially overlap with each other, we assume the decision of the expert to them are independent. Therefore, in Eq. 6, the probability in the entropy is calculated by:

$$P(Y|X;\theta_{\mathcal{T}}) = \prod_i P(l_i|\mathbf{x}_i;\theta_{\mathcal{T}}). \tag{7}$$

The entropy regularization term describes the degree of ambiguity of the tracker for the target candidate set. By adding the entropy prior, the score favors the expert with less ambiguity with respect to the possible label set Z. We state that the trajectory consistency plays the main role in the whole score $\mathcal{S}_{\mathcal{T}}$, and the entropy is the sufficient complement to it.

3.2. Correlation Tracking

Recently, correlation filter based trackers have draw much attention due to its high efficiency and robustness. And correlation trackers have show their outstanding performance in public evaluation dataset and benchmark [16, 27]. For the accuracy and low computational cost purpose, we train the correlation filter as the base tracker in our framework. We use the ridge regression model to learn the correlation of the temporal target context [12, 11]. In addition, by taking all the circular shift of image patches into consideration, the model produces less ambiguous response map than the binary classifier, which is more suitable to our tracking framework.

Correlation filter tracker models the appearance of the target on an extended $M \times N$ image patch $\mathbf{x}$ which is centered by the target position. The goal is to find a classifier $f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle$ to make prediction for the probability of image patch. Instead of sampling image patch with step, the classifier is trained with all the circular shift of $\mathbf{x}_i$, where $i \in \{0, ..., M-1\} \times \{0, ..., N-1\}$. Each training example $\mathbf{x}_i$ is assigned a training label generated by Gaussian function y_i. The training goal is to minimize the regression error:

$$\min_{\mathbf{w}} \sum_i (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - y_i)^2 + \lambda \|\mathbf{w}\|^2, \tag{8}$$

where ϕ is the mapping to the kernel space, and λ is the regularization parameter that controls overfitting. With kernel trick, $\mathbf{w}$ can be denoted by a linear combination of the training samples: $\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x}_i)$, where $\boldsymbol{\alpha}$ is the dual space coefficients of $\mathbf{w}$. Given the kernel defined by $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, the classifier is derived by $f(\mathbf{x}) = \sum_i \alpha_i \kappa(\mathbf{x}_i, \mathbf{x})$. Then the optimization problem is transformed under $\boldsymbol{\alpha}$ instead of $\mathbf{w}$. Let the hat symbol "$\hat{}$" denotes the Discrete Fourier Transform (DFT). According to [12], for a unitarily invariant kernel, $\boldsymbol{\alpha}$ is derived as:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda}, \tag{9}$$

where $\hat{\mathbf{k}}^{\mathbf{xx}}$ is the so-called kernel correlation whose i-th element is $\kappa(\mathbf{x}_i, \mathbf{x})$. The kernel correlation can also be computed efficiently in the Fourier domain. Particularly, for the Gaussian kernel, when the input patch $\mathbf{x}$ has multiple channels, which is concatenated by individual vectors of C channels, i.e. $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_C]$, the kernel correlation can be computed by:

$$\mathbf{k}^{\mathbf{xx}'} = \exp(-\frac{1}{\sigma^2}(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathcal{F}^{-1}(\sum_c \hat{\mathbf{x}}_c^* \odot \hat{\mathbf{x}}_c'))), \tag{10}$$

where $\mathcal{F}^{-1}$ is the Inverse DFT (IDFT), and $\odot$ denotes the element-wise product.

After the above training procedure, the detection task is carried out on an image patch $\mathbf{z}$ in the new frame within the $M \times N$ window, which is centered at the last target position. The response map can be evaluated by:

$$f(\mathbf{z}) = \mathcal{F}^{-1}(\hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\boldsymbol{\alpha}}). \tag{11}$$

Therefore, the new position of the target is determined by the maximum of $f(\mathbf{z})$. To move discontinuities at the image boundaries of the response map, the input feature channels are weighted by a cosine window [4]. To adapt to the appearance change of the target, the linear interpolation strategy is conducted on $\boldsymbol{\alpha}$ and $\mathbf{x}$:

$$\hat{\boldsymbol{\alpha}}^t = (1-\gamma)\hat{\boldsymbol{\alpha}}^{t-1} + \gamma\hat{\boldsymbol{\alpha}}, \tag{12}$$

$$\hat{\mathbf{x}}^t = (1-\gamma)\hat{\mathbf{x}}^{t-1} + \gamma\hat{\mathbf{x}}, \tag{13}$$

where γ is the learning rate.

In our tracker, we employ the PCA-HOG feature described in [12]. Besides the HOG feature which puts more emphasis on the object shape, we further add the color feature to promote the tracker performance. Here we apply color attribute feature to map the RGB values to the probabilistic 11 dimensional color representation [26, 7].

In order to deal with the scale variation of the target, we adopt a scale adaptive method to our tracker. Unlike [6, 22], our method estimates the translation and target scale simultaneously. Let $M_t \times N_t$ denotes the search window size at time t, we first establish a target pyramid through cropping image patches, all of which are centered at the target position of time $t-1$, each of size $(1+as)M_t \times (1+as)N_t$, where a is a constant scalar of the scale factor, and $s \in \{\lfloor -\frac{N_s-1}{2} \rfloor, \lfloor -\frac{N_s-3}{2} \rfloor, ..., \lfloor \frac{N_s-1}{2} \rfloor\}$ is the scale index. Then all the N_s image patches are resized to the target template size. After that, the response map of each cropped image patch can be evaluated according to Eq. 11, all of which

constitute the response pyramid. Finally, the accurate scale index is indicated by the maximum of the response pyramid, as well as the translation (which should multiply by its ratio relative to the template size).

3.3. Scale-adaptive Multi-Expert (SME) Tracker

Given the above expert selection criteria and the base tracker of correlation filter, we propose our tracker, named Scale-adaptive Multi-Expert (SME).

The snapshots in the expert ensemble are stored chronologically at intervals of Ω frames. When the number of experts exceeds the maximum number $N_{\mathbf{E}}$, the oldest expert is discarded. At each frame, each expert in the ensemble gets its own decision of the target position by calculating the maximum value of the correlation filter response map. In addition, the expert ensemble proposes the potential target candidates X. After that, expert scores are calculated each frame by Eq. 5. Whenever there is a disagreement among the experts, the best expert is selected according to their accumulative score define by Eq. 1, and displaces the current expert to be the current tracker. Otherwise, the target is tracked by the current expert $\mathcal{T}_T$. Note that only expert $\mathcal{T}_T$ is updated, so the whole algorithm has low computation.

At the same time, target scale is estimated according to the method described in Section 3.2. Generally, scale variation is much smaller than that of translation. For computation efficiency, the scale estimation is only conducted on the current expert $\mathcal{T}_T$. We find this strategy very effective in practice.

4. Implementation

The whole algorithm procedures of our SME tracker is shown in the following Algorithm flowchart. Some implementation details are discussed below.

The target position used for calculating the trajectory consistency score is given by the global maxima of the response map of each expert. The target positions decided by each expert are then processed by hierarchical clustering according to their spatial distribution. If the clustering result has more than one class, a disagreement is reported. In order to obtain the target candidate set X, we first get all the samples proposed by every expert whose response values are greater than the pre-defined threshold ε. And then merge the samples at their mean position if their distance is smaller than δ to avoid heavily overlap. The sample probability $P(+1|\mathbf{x}_i;\theta_{\mathcal{T}})$ in Eq. 7 is naturally obtained by the response map of the base tracker, and $P(-1|\mathbf{x}_i;\theta_{\mathcal{T}})$ is got by $1-P(+1|\mathbf{x}_i;\theta_{\mathcal{T}})$.

The parameters setup are as follows. The max number of experts $N_{\mathbf{E}}$ is set to 4. The window size Δ and frame interval Ω are set to 4 and 50 respectively. Let the template target size denoted by l. We set the cutoff distance of the hierarchical clustering and merge threshold δ equally by $l/2$. The parameter σ of trajectory consistency score in Eq. 3 is $l/3$, which is set according to the 3-sigma rule of Gaussian distribution. Note that all the response maps of the expert ensemble (including each layer of response pyramid generated by the current expert) are of the same template size. Therefore the above parameters are not influenced by target size. The tradeoff parameter in Eq. 5 is $\eta = 15$, and the candidate selection threshold $\varepsilon = 0.8$.

For the base tracker, we adopt the Gaussian kernel $\kappa(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2})$, and its kernel width is set to 0.5. The regularization parameter λ in Eq. 8 is set to 10^{-4}. The padding size and learning rate γ are set to 1.8 and 0.01 respectively. The number of target pyramid layers N_s is 9, and the scale factor a is 0.005. The template size is set to the initial target size.

Algorithm 1: SME Tracker

input : Initial target bounding box $\mathbf{x}_1$
output: The estimated target state $\mathbf{x}_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$

$\mathbf{E} \leftarrow \mathcal{T}_1$
repeat
  Get the target candidate set X by $\mathbf{E}$;
  for $\mathcal{T} \in \boldsymbol{E}$ **do**
    if $\mathcal{T} = \mathcal{T}_t$ **then**
      Build the target pyramid at $(\hat{x}_{t-1}, \hat{y}_{t-1})$;
      Get the response pyramid, estimate the target position $(x_{\mathcal{T}_t}.y_{\mathcal{T}_t})$ and scale $\hat{s}_t$;
    else
      Get the response map and estimate the target position $(x_{\mathcal{T}}, y_{\mathcal{T}})$;
    Compute the expert score $\mathcal{S}^t_{\mathcal{T}}$;
  if *expert disagreement is reported* **then**
    Select $\mathcal{T}^* \in \mathbf{E}$ according to Eq. 1;
    $\mathbf{x}_t = (x_{\mathcal{T}^*_t}, y_{\mathcal{T}^*_t}, \hat{s}_t)$;
    $\mathcal{T}_t \leftarrow \mathcal{T}^*$;
  else
    $\mathbf{x}_t = (x_{\mathcal{T}_t}, y_{\mathcal{T}_t}, \hat{s}_t)$;
  if $\mod(t, \Omega) == 0$ **then**
    $\mathbf{E} \leftarrow \mathbf{E} \cup \mathcal{T}_t$;
    discard the oldest snapshot when $|\mathbf{E}| > N_{\mathbf{E}}$;
  Update $\mathcal{T}_t$;
until *Last frame of video sequences*;

5. Experiments

In this section, we evaluate our SME tracker on two large dataset, one is the 51 sequences Visual Tracker Benchmark [27], the other is the 60 sequences VOT2015 Challenge dataset. On the former dataset, we compare the proposed tracker with state-of-the-art trackers to demonstrate its excellent performance. Then, the tracker is tested on

sequences of eight main attributes to analysis the performance of SME in different scenarios. We also decompose SME into different parts to analysis the effectiveness of the proposed framework. To further verify the efficiency of our tracker, we test SME on the new VOT2015 dataset, which contains 60 sequences, the experimental result of the VOT dataset is reported for evaluation. The implementation and more experimental results are publicly available[1].

5.1. Experiment Setup

The proposed SME tracker is implemented in Matlab&C. Although with multiple experts, our tracker runs at roughly 37.5fps on the 3.20GHz CPU with 8GB RAM, mostly due to the efficiency of the correlation filter. The parameters setup is in accordance to the description in Section 4.

In the 51 sequences Visual Tracker Benchmark, the quantitative analysis is illustrated on two evaluation plots: (i) the success plot and (ii) the precision plot. The success plot is based on the bounding box overlap metric, and shows the percentage of successful frames at the overlap threshold varies from 0 to 1. The ranking is according to the area under curve (AUC) score. The precision plot shows the ratio of frames whose center location error (CLE) is within a given threshold.

The VOT2015 dataset contains 60 short challenging sequences. The sequences are annotated using rotated bounding box in order to provide highly accurate ground truth, which is different from the 51 sequences benchmark annotated by rectangles. The VOT2015 dataset is evaluated by two criteria: (i) accuracy and (ii) robustness. The accuracy measures the overlap between the tracking result and ground truth. The robustness measures how many times the tracker loses the target.

5.2. Visual Tracker Benchmark

Overall performance. Besides the 29 trackers provided by [27], we add four recently state-of-the-art trackers including KCF [12], CN [7], MEEM [29], and TGPR [8]. According to the evaluation methods by Visual Tracker Benchmark, the one-pass evaluation (OPE) performance is illustrated in the Success Plot and Precision Plot shown in Fig. 2.

For clear illustration, we plot the top-10 among all the compared trackers. As shown in the plots, our tracker achieves 0.628 success score and 0.836 precision score, both of which rank first among all the trackers. Particularly, MEEM is most similar to our tracker. Compared to MEEM, SME surpasses it with large margin, especially exceeds in the success score by 11.5%. KCF is the also the correlation filter based method, which can be regard as our base tracker. Compared to KCF, our tracker improves the overlap success and precision score by 22.2% and 12.97% respectively. The overall plots demonstrate our tracker is effective and promising.

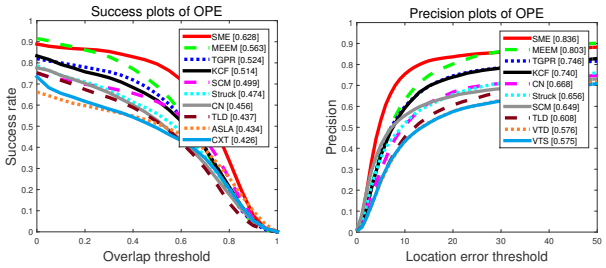

Figure 2: The success plot and precision plot over 51 sequences Visual Tracker Benchmark using one pass evaluation (OPE). The legend illustrates the area under curve (AUC) for the success plot, and the score of the threshold 20 for the precision plot.

Attribute-based performance. In this experiment, the benchmark sequences are divided into 8 main attributes to evaluate the tracker in different scenarios. As described in [27], the AUC score of success plot measures the tracker performance more accurate than precision plot of one threshold, so the success plot is the main analysis evaluation. Therefore, we report the eight main attributes of success plots in Fig. 3. As illustrated in the plots, SME ranks first in all the attributes.

SCM shows high score of 0.518 in the scale variation attribute, while SME performs better with 0.585 success scores. The MEEM performs well with 0.557 points in background clutter, 0.560 in motion blur, and 0.647 in out-of-view scenario. While SME shows more preferable performance in all these scenarios. Particularly, in the attributes of scale variation, occlusion, out-of-plane rotation, deformation and illumination variation, SME exceeds the second rank tracker by around 10%. In detail, SME has improved all the attributes by $13.3\%, 17.6\%, 9.7\%, 5.0\%, 0.7\%, 9.3\%, 13.1\%$ and 0.15% respectively compared to the second rank tracker. Among all the attributes, the scale variation performance is improved significantly, which shows our scale scheme is very effective. In addition, SME also gets more favorable scores than other correlation filter based trackers, KCF [12] and CN [7], which demonstrates the effectiveness of the multi-expert framework.

Component analysis. To further demonstrate the effectiveness of the proposed tracker, we decompose our approach into two trackers with part of the features of the original SME: (i) SME-base, the base correlation tracker. (ii) SME-sfix, the original SME without scale estimation. We summarize their success and precision score in Table. 1.

[1] https://sites.google.com/site/jiatonglihome/research/sme-tracker

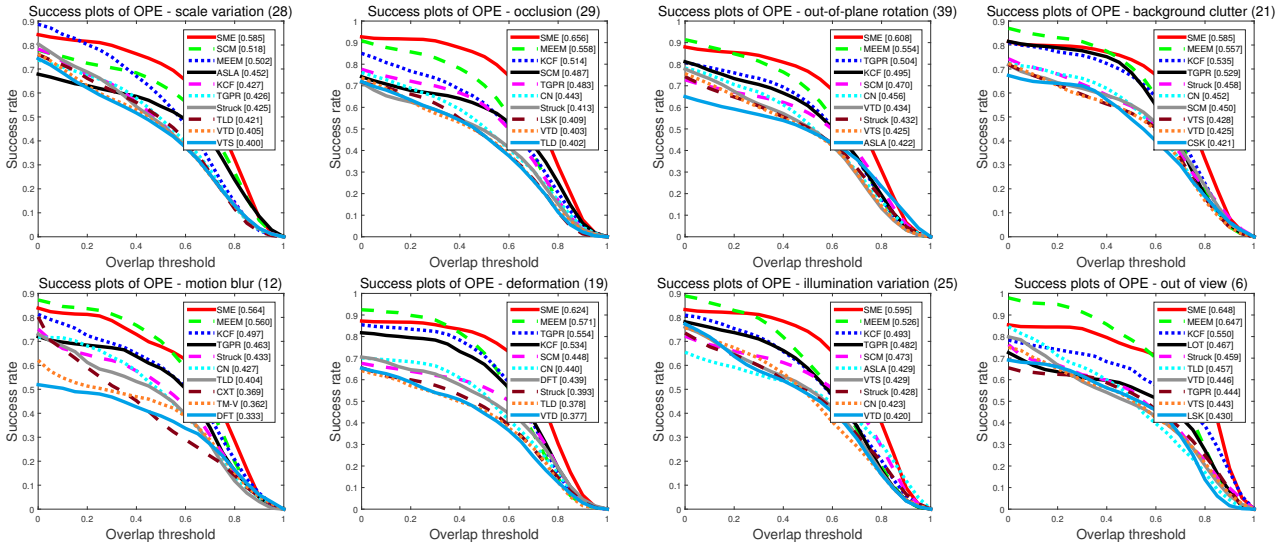

Figure 3: The success plots of eight attributes of the benchmark, i.e. scale variation, occlusion, out-of-plane rotation, background clutter, motion blur, deformation, illumination variation and out of view. The legend illustrates the AUC score for each tracker.

Table 1: Success and Precision Score of the SME Analysis

	SME	SME-base	SME-sfix
Success	0.628	0.608	0.598
Precision	0.836	0.792	0.812

From the table, both the success and precision score of SME are higher than its counterparts. Moreover, the precision score of SME-base decreases from 0.836 to 0.792 compared to original SME, which means that the multi-experts framework is important in improving the CLE score in tracking. On the other hand, when remove the scale estimation, the success score of SME-sfix declines immediately. However, the precision of SME-sfix falls relatively smaller compared to SME-base. It is reasonable since the multi-expert framework pays more attention to correct the estimation error of target translation by selecting reliable historical tracker snapshots, while the scale estimation is more related to overlap metric. Therefore, the combination of the two gives satisfactory effect.

Finally, some of the typical tracking results from the top 7 trackers are shown in Fig. 4, including SME, TGPR [8], MEEM [29], KCF [12], CN [7], SCM [25] and Struck [10]. The sequences are *Singer1, Soccer, Dog1, Jogging, Skating1, Bolt, Trellis and Walking2*. Among all the test sequences, *Singer1* and *Dog1* have significant scale changes, and *Soccer*, *Jogging*, *Skating1* and *Walking2* go through part or whole occlusion. In addition, *Singer1* and *Skating1* also have illumination variation due to the stage light, as well as *Trellis* due to the sunshine. From *Singer1* and *Dog1*, we can see that SME performs well in handling scale variation. Especially in *Dog1* with large scale change in frame 1046 and frame 1275, the proposed algorithm gives accurate scale estimation compared to SCM. In *Soccer*, where most of the compared trackers fail, SME is able to catch the target despite of its significant background clutter. This is because our tracker combines both the HOG and color feature, so as to adapt to the target blur caused by background clutter. When there is large object appearance changes, like in frame 412 of *Trellis*, most of the compared algorithms start to drift, but our tracker is capable to deal with the challenge. The same phenomenon can be found in *Bolt*. In the *Walking2* sequence, when another pedestrian with same appearance appears in frame 262, CN, MEEM and TGPR cannot distinguish between them and start to track the distractor, however, our tracker with multi-expert can handle this scenario well.

5.3. VOT2015 Challenge Dataset

The number of sequences in VOT2015 Challenge Dataset has been enlarged to 60 compared to VOT2013 and VOT2014, whose numbers of sequences are 16 and 25 respectively. On this dataset, we compare the performance of SME with three trackers, DSST [6], MEEM [29] and KCF [12]. MEEM, TGPR [8] and KCF are the top-3 trackers besides SME in the 51 sequences benchmark. TGPR is not compared in this dataset because of its high computa-

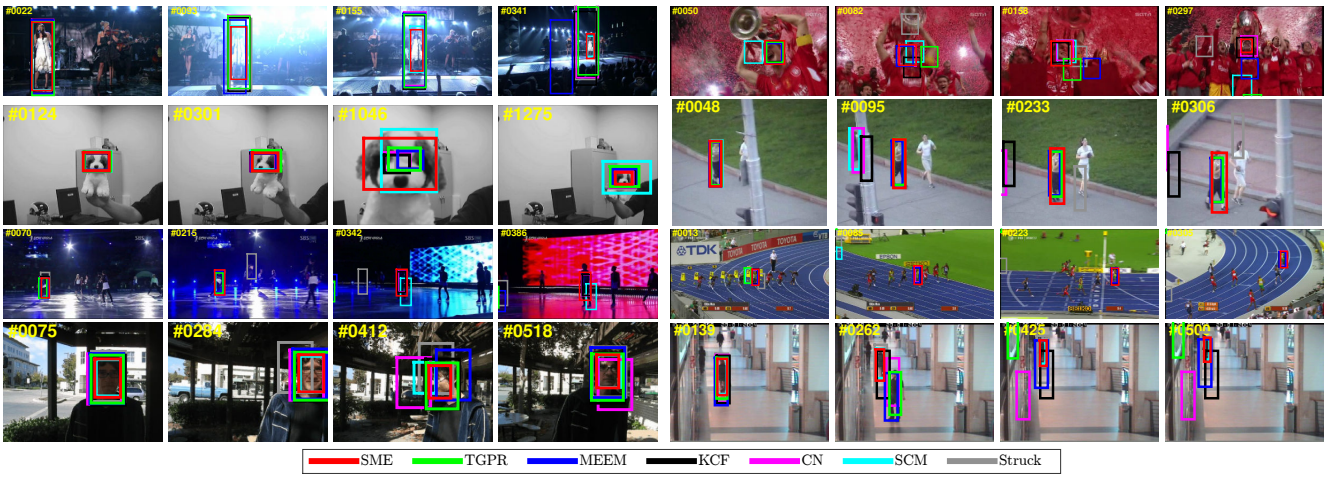

Figure 4: Tracking results of the top seven algorithms (TGPR [8], MEEM [29], KCF [12], CN [7], SCM [25] and Struck [10]) in Fig. 2 over eight sequences. The illustration example videos from top-left to bottom-right are *Singer1, Soccer, Dog1, Jogging, Skating1, Bolt, Trellis and Walking2*.

Table 2: The results of VOT2015 Challenge Dataset.

	A-Rank	R-Rank	Accuracy	Robustness
SME	2.00	2.27	0.50	1.98
DSST	2.41	2.65	0.49	2.31
MEEM	2.70	2.47	0.44	2.41
KCF	2.91	2.61	0.43	2.53

[2] red: rank 1, blue: rank 2, green: rank 3

tion cost . Since DSST is the winner of the VOT2014 challenge [16], the comparison with it can validate the performance of our tracker to a large extent. In addition, MEEM is close to our work. To verify the superiority of SME to MEEM, we choose to compare with it further in this larger dataset.

According to the VOT evaluation criteria [16], the overall experimental results are illustrated in two plots: (i) accuracy-robustness (AR) ranking plot and (ii) AR plot, as shown in Fig. 5, and the AR ranking plot is the main evaluation criteria. The AR ranking plot shows average ranking score of all the sequences for each tracker in the joint accuracy-robustness rank space. The AR plot is the data visualization shows the average accuracy-robustness data of each tracker. For both plots, the tracker is better if the legend resides closer to the top-right corner of the plots. The details about the evaluation method is referred to [16, 5].

From the plots, it is indicated that SME ranks higher than all the other compared trackers. DSST ranks second and MEEM ranks third. The data in the plots are listed in Table. 2, which also demonstrates the excellent performance of the proposed tracker.

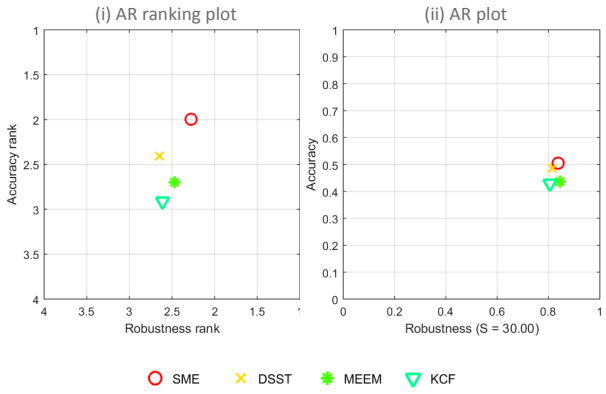

Figure 5: The AR ranking plot and the AR plot for VOT2015 Challenge Dataset. The tracker is better if its legend resides closer to the top-right corner of the plot. S is the data visualization parameter.

6. Conclusions

In this paper, we propose an effective scale adaptive multi-expert tracker. The multi-expert is composed of both the current tracker and its historical snapshots. The best expert is selected by the proposed trajectory consistency score. Each expert is learned by the discriminative correlation filter, while the scale is estimated by searching the target pyramid. The experiments are conducted on two large tracking datasets, which demonstrate the proposed tracker performs favorably against state-of-the-art methods.

Acknowledgment

This project is supported by International Graduate Exchange Program of Beijing Institute of Technology.

References

[1] S. Avidan. Support vector tracking. *TPAMI*, 26(8):1064–1072, 2004.

[2] S. Avidan. Ensemble tracking. *TPAMI*, 29(2):261–271, 2007.

[3] B. Babenko, M. H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *TPAMI*, 33(8):1619–1632, 2011.

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

[5] L. Cehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? *In WACV*, pages 540–547, 2014.

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

[7] M. Danelljan, F. S. Khan, M. Felsberg, and van de Weijer. Adaptive color attributes for real-time visual tracking. *In CVPR*, pages 1090–1097, 2014.

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

[9] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *In NIPS*, pages 529–536, 2004.

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

[11] 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, 2012.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[13] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. *In CVPR*, pages 749–758, 2015.

[14] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. *In ECCV*, pages 155–171, 2014.

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

[16] M. Kristan, R. Pflugfelder, R. Leonardis, et al. The visual object tracking vot2014 challenge results. *In ECCV Workshop*, pages 191–217, 2014.

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

[18] J. Kwon and K. M. Lee. Tracking by sampling and integratingmultiple trackers. *TPAMI*, 36(7):1428–1441, 2014.

[19] D. Y. Lee, J. Y. Sim, and C. S. Kim. Multihypothesis trajectory analysis for robust visual tracking. *In CVPR*, pages 5088–5096, 2015.

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

[21] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. *In ECCV Workshop*, pages 254–265, 2014.

[22] C. Ma, X. Yang, C. Zhang, and J. M. H. Yang. Long-term correlation tracking. *In CVPR*, pages 5388–5396, 2015.

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

[24] N. Wang, S. Li, A. Gupta, and D. Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.

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

[26] J. V. D. Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 18(7):1512–1523, 2009.

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

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

[29] J. Zhang, S. Ma, and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. *In CVPR*, 2014.

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