End-to-end Flow Correlation Tracking with Spatial-temporal Attention

Zheng Zhu, Wei Wu, Wei Zou and Junjie Yan
Institute of Automation, CAS
SenseTime Group Limited

Abstract

Discriminative correlation filters (DCF) with deep convolutional features have achieved favorable performance in recent tracking benchmarks. However, most of existing DCF trackers only consider appearance features of current frame, and hardly benefit from motion and interframe information. The lack of temporal information degrades the tracking performance during challenges such as partial occlusion and deformation. In this work, we focus on making using of the rich flow information in consecutive frames to improve the feature representation and the tracking accuracy. The historical feature maps is warped and aggregated with current ones by guiding of flow and an end-to-end training framework is developed for tracking. Specifically, individual components, including optical flow estimation, feature extraction, aggregation and correlation filter tracking are formulated as special layers in network. Then the previous frames at predefined intervals are warped to the current frame using optical flow information. Meanwhile, we propose a novel spatial-temporal attention mechanism to adaptively aggregate warped feature maps as well as current features maps. All the modules are trained end-to-end. To the best of our knowledge, this is the first work to jointly train flow and tracking task in a deep learning framework. Extensive experiments are performed on four challenging tracking datasets: OTB2013, OTB2015, VOT2015 and VOT2016, and our method achieves superior results on these benchmarks.

1. Introduction

Visual object tracking, which tracks a specified target in a changing video sequence automatically, is a fundamental problem in many topics such as visual analytics [1], automatic driving [2], pose estimation [3] and et al. A core problem of tracking is how to detect and locate the object accurately in changing scenarios with occlusions, shape deformation, illumination variations and et al [4, 7].

Recently, significant attention has been paid to discriminative correlation filters (DCFs) based methods [9, 10, 11, 17, 27, 46, 44, 14, 15, 19, 20, 24, 43] for visual tracking. The DCF trackers can efficiently train a repressor by exploiting the properties of circular correlation and performing operations in the Fourier domain. Except for conventional DCF trackers [9, 17], Many improvements for DCF tracking approaches have also been proposed such as SAMF, LCT [11] MUSTer [46], SRDCF [10] and CACF [44]. Inspired by the success of CNN in object classification [12, 13], detection [14] and segmentation tasks[15], the visual tracking community has started to focus on the deep trackers that exploit the strength of convolution neural networks (CNNs) in recent years. HCF [19] and HDT [20] combines hierarchical response and hedge weak trackers to abtain tracking results, respectively. DeepSRDCF [24] exploits shallow CNN features in a spatially regularized DCF framework. CFNet [43] unifies correlation filters and similarities learning into a siamese tracking framework. However, most existing DCF trackers only consider appearance features of current frame, and are hardly benefit from motion and interframe information. The lack of temporal information degrades the tracking performance during chal-

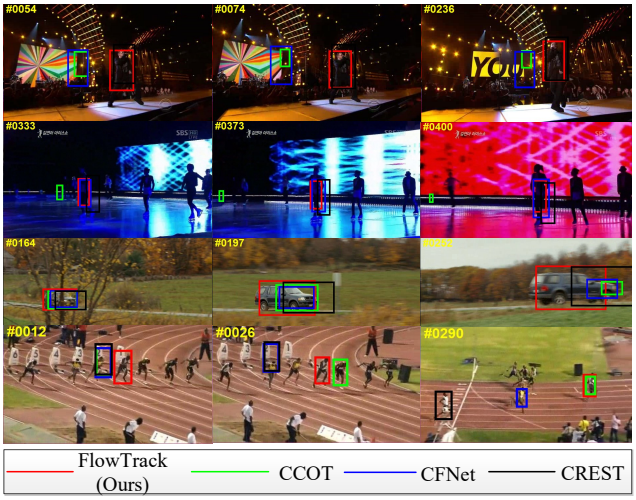

Figure 1: Comparisons of our approach with three state-of-the-art trackers in the challenging scenario.

lenges such as partial occlusion and deformation.

Optical flow encodes correspondences between two input images. It is usually obtained by variational approaches [35, 36] and combinatorial matching[37, 38]. Recent study focuses on directly appling deep CNNs to estimate the motion, such as FlowNet [56] and FlowNet2.0 [57]. Flow information has been exploited to be helpful in computer vision tasks, such as pose estimation [49], frame prediction [50], and attribute transfer [51]. Although some trackers utilize optical flow to upgrade performance[54, 55], the flow feature is off-the-shelf and not trained end-to-end. These methods do not take full advantage of flow information, so achieved performance may be suboptimal. Note that the features of the same object instance are usually not spatially aligned across frames due to video motion. A naive feature fusion may even deteriorate the performance. This suggests that it is critical to model the motion during learning.

In this paper, we develop an end-to-end framework for tracking which utilizing both the flow information and appearance features. Specifically, the previous frames are warped to current frame by guiding of flow information and then they are aggregated for consequent correlation filter tracking. In order to adaptively weight warped feature maps and current feature maps, a novel spatial-temporal attention mechanism is developed. All the modules, including optical flow estimation, feature extraction and aggregation, correlation filter tracking are trained end-to-end.

1.1. Contributions

The contributions of this paper can be summarized in three folds as follows:

1, We develop an end-to-end tracking framework that aggregates historical feature maps with current ones using flow information to improve the feature representation and the tracking accuracy. To the best of our knowledge, this is the first work to jointly train flow and tracking task in a deep learning framework.

2, A novel spatial-temporal attention mechanism is proposed for adaptive aggregation. In spatial attention, feature maps are weighted in planar position to encode spatial similarities. Then, channels of feature maps are further reweighted to take temporal attention into account.

3, Extensive experiments are carry out on tracking benchmarks and demonstrate that the proposed tracking algorithm performs favorably against existing state-of-the-art methods in terms of accuracy and robustness. Figure 1 shows a comparison to state-of-the-art trackers on four benchmark sequences.

2. Related works

Visual tracking is a significant problem in computer vision systems and a series of approaches have been proposed in recent years. Since our main contribution is an end-to-end framework for flow correlation filters tracking, we give a brief review on three directions closely related to this work: DCF-based trackers,CNN-based trackers, and optical flow in visual recognition.

2.1. DCF with hand-crafted features

In recent tracking community, significant attention has been paid to discriminative correlation filters (DCF) based methods [9, 10, 11, 27, 16, 17, 31, 45, 44, 60, 46, 47, 59] beuause of their efficiency and expansibility. Correlation operations can be performed by element-wise multiplications using discrete Fourier transform in frequency domain, and discrete Fourier transform can be computed by the efficient fast Fourier transform in practice. This property of correlation filters makes it efficient for fast training and detection in visual tracking. In 2010, the Minimum Output Sum of Squared Error (MOSSE) [16] filter is proposed , which is the first correlation filter-based tracking method. Then, Henriques et al. [9, 17] proposed the Kernelized Correlation Filters (KCF), which is the improvement of the MOSSE filter by introducing kernel methods and extending it to deal with multiple channels. Many improvements for DCF tracking approaches have also been proposed, such as SAMF [27] and fDSST[31] for scale changes, CN [39] and Staple [18] taking color information into account, LCT [11] and MUSTer [46] for long-term tracking, SRDCF [10], CSR-DCF [59], CACF [44] and BACF [60] to mitigate boundary effects, SCT [47] and MCPF [45] for multiple template or multi-task. The better performance is obtained, the more time DCF based tracker cost. Most these methods use handcrafted features, which hinder their accuracy and robustness. It is noticed that increasing research about deep learning is introduced in correlation tracking community, which will be described in the next subsection.

2.2. DCF with CNN features

Inspired by the success of CNN in object recognition [12, 13, 14], researchers in tracking community have started to focus on the deep trackers that exploit the strength of CNN. Since DCF provides an excellent framework for recent tracking research, the popular trend is the combination of DCF framework and CNN features. In HCF [19] and HDT [20], the CNN are employed to extract features instead of handcrafted features, and final tracking results are obtained by combining hierarchical response and hedging weak trackers, respectively. DeepSRDCF [24] exploits shallow CNN features in a spatially regularized DCF framework. In above mentioned methods, the chosen CNN features are always pre-trained in different task and individual components in tracking systems are learned separately. So the achieved tracking performance may be suboptimal. It is worth noting that CFNet [43] interprets the correlation

filters as a differentiable layer in a Siamese tracking framework, thus achieving an end-to-end representation learning. The main drawback is its unsatisfying performance.

2.3. CNN-based trackers

Except for the combination of DCF framework and CNN features, another trend in deep trackers is to design the tracking networks and pre-train them in order to learn the target-specific features and handle the challenges for each new video. Bertinetto et.al [25] propose a fully convolutional siamese network (SiamFC) to estimate the feature similarity region-wise between two frames. The network is trained off-line and evaluated without any online fine-tuning. Similar to SiamFC, in GOTURN tracker [26], the motion between successive frames is predicted using a deep regression network. MDNet [21] trains a small-scale network by multi-domain methods, thus separating domain independent information from domain-specific layers. C-COT [22] and ECO [23] employ the implicit interpolation method to solve the learning problem in the continuous spatial domain, where ECO is an improved version of C-COT in performance and speed. CREST [62] treats tracking process as convolution and apply residual learning to take appearance changes into account. Similarly, UCT [61] treats feature extractor and tracking process both as convolution operation and trains them jointly, enabling learned CNN features are tightly coupled to tracking process. All these trackers only consider appearance features in current frame and are hardly benefit from motion and interframe information. In this paper, we make full use of these information by aggregating flow and Siamese tracking in an end-to-end framework.

2.4. Optical flow for visual recognition

Flow information has been exploited to be helpful in computer vision tasks. In pose estimation [49], optical flow is used to align heatmap predictions from neighbouring frames. [50] applies flow to the current frame to predict the next frame. In [51], flow is used to explicitly model how image attributes vary with its deformation. DFF [52] and FGFA [53] utilize flow information to speed up vision recognition (segmentation and video detection) and upgrade performance, respectively. In DFF, expensive convolutional sub-network is performed only on sparse key frames, and their deep feature maps is propagated to other frames via a flow field. In FGFA, nearby features are aggregated along the motion paths using flow information, thus improving the video recognition accuracy. Recently, some trackers also utilize optical flow to upgrade performance[54, 55], while the flow feature is off-the-shelf and not trained end-to-end. Since the features of the same object instance are usually not spatially aligned across frames due to video motion, a naive feature fusion may not gain performance.

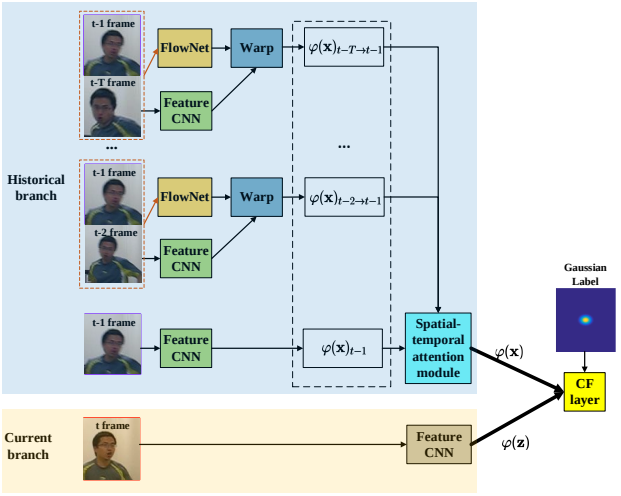

Figure 2: The overall training network architecture.

3. End-to-end learning of flow correlation tracking

In this section, flow correlation network is given at first to describe the overall training architecture. Then we introduce the correlation filter layer and the aggregation of optical flow. In order to adaptively weight the importance of aggregated frames at each spatial location and temporal channels, a novel spatial-temporal attention mechanism is designed at last.

3.1. Overall training network architecture

The overall training framework of our tracker consists of feature extraction sub-network, FlowNet sub-network, warping module, spatial-temporal attention sub-network and DCF tracking module. As shown in Figure 2, overall training architecture adopts Siamese network. Firstly, appearance features and flow information are extracted by feature CNN and FlowNet, respectively. Then previous frames at predefined intervals (5 frames in experiments) is warped to current frame guided by flow information. Meanwhile, a spatial-temporal attention module is designed to weight the warped feature maps as well as current frame's. Finally, the learning and detection branch is fed into subsequent DCF layer for training. All the modules are differentiable and trained end-to-end.

3.2. Correlation filter layer

Discriminative correlation filters (DCF) with deep convolutional features have shown favorable performance in recent benchmarks [19, 20, 24]. Nonetheless, the chosen CNN features are always pre-trained in different task and individual components in tracking systems are learned separately, thus the achieved tracking performance may be sub-optimal. Recently, CFNet [43] and DCFNet [33] interprets the correlation filters as a differentiable layer in a Siamese

tracking framework, thus achieving an end-to-end representation learning.

In DCF tracking framework, the aim is to learn a series of convolution filters $\mathbf{f}$ from training samples $(\mathbf{x}_k, \mathbf{y}_k)_{k=1:t}$. Each sample is extracted using another CNN from an image region. Assuming sample has the spatial size $M \times N$, the output has the spatial size $m \times n$ ($m = M/stride_M, n = N/stride_N$). The desired output $\mathbf{y}_k$ is a response map which includes a target score for each location in the sample $\mathbf{x}_k$. The response of the filter on sample $\mathbf{x}$ is given by

$$R(\mathbf{x}) = \sum_{l=1}^{d} \varphi^l(\mathbf{x}) * \mathbf{f}^l \tag{1}$$

where $\varphi^l(\mathbf{x})$ and $\mathbf{f}^l$ is l-th channel of extracted CNN features and desired filters, respectively, $*$ denotes circular correlation operation. The filter can be trained by minimizing L_2 loss which is obtained between the response $R(\mathbf{x}_k)$ on sample $\mathbf{x}_k$ and the corresponding Gaussian label $\mathbf{y}_k$

$$L = ||R(\mathbf{x}_k) - \mathbf{y}_k||^2 + \lambda \sum_{l=1}^{d} ||\mathbf{f}^l||^2 \tag{2}$$

The second term in (2) is a regularization with a weight parameter λ. The solution can be gained as [33]:

$$\mathbf{f}^l = \mathcal{F}^{-1}\left(\frac{\hat{\varphi}^l(\mathbf{x}) \odot \hat{\mathbf{y}}^*}{\sum_{l=1}^{d} \hat{\varphi}^l(\mathbf{x}) \odot (\hat{\varphi}^l(\mathbf{x}))^* + \lambda}\right) \tag{3}$$

where the hat symbol represents the discrete Fourier transform of according variables, * represents the complex conjugate of according number, and $\odot$ denotes Hadamard product.

In test stage, the trained filters are used to evaluate an image patch centered around the predicted target location:

$$R(\mathbf{z}) = \sum_{l=1}^{d} \varphi^l(\mathbf{z}) * \mathbf{f}^l \tag{4}$$

where $\varphi(\mathbf{z})$ denote the feature map extracted from tracked target position of last frame including context.

In order to unify the correlation filters in an end-to-end network, we formulate above solution as CF layer. Given the feature maps of search patch $\varphi^l(\mathbf{z})$, the loss function is formulated as:

$$\begin{aligned} L(\boldsymbol{\theta}) =& ||R(\boldsymbol{\theta}) - \tilde{R}||^2 + \gamma ||\boldsymbol{\theta}||^2 \\ s.t. \quad R(\boldsymbol{\theta}) =& \sum_{l=1}^{d} \varphi^l(\mathbf{z}, \boldsymbol{\theta}) * \mathbf{f}^l \\ \mathbf{f}^l =& \mathcal{F}^{-1}\left(\frac{\hat{\varphi}^l(\mathbf{x}, \boldsymbol{\theta}) \odot \hat{\mathbf{y}}^*}{\sum_{l=1}^{d} \hat{\varphi}^l(\mathbf{x}, \boldsymbol{\theta}) \odot (\hat{\varphi}^l(\mathbf{x}, \boldsymbol{\theta}))^* + \gamma}\right) \end{aligned} \tag{5}$$

where $\tilde{R}$ is desired response, and it is a gaussian distribution centered at the real location. The backpropagation of loss with respect to $\varphi(\mathbf{x})$ and $\varphi(\mathbf{z})$ are listed as follows respectively [33]:

$$\begin{aligned} \frac{\partial L}{\partial \varphi^l(\mathbf{x})} &= \mathcal{F}^{-1}\left(\frac{\partial L}{\partial (\hat{\varphi}^l(\mathbf{x}))^*} + \left(\frac{\partial L}{\partial (\hat{\varphi}^l(\mathbf{x}))}\right)^*\right) \\ \frac{\partial L}{\partial \varphi(\mathbf{z})^l} &= \mathcal{F}^{-1}\left(\frac{\partial L}{\partial (\hat{\varphi(\mathbf{z})}^l)^*}\right) \end{aligned} \tag{6}$$

Once the backpropagation is derived, the correlation filters can be formulated as a layer in network, which is called correlation filter layer in next sections.

3.3. Aggregation using optical flow

Optical flow encodes correspondences between two input images. We warp the feature maps from the neighbor frames to current frames according to the flow:

$$\varphi_{t \to i} = \mathcal{W}(\varphi_t, Flow(I_t, I_i)) \tag{7}$$

where $\varphi_{t \to i}$ denotes the feature maps warped from previous frame t to current frame i. $Flow(I_t, I_i)$ is the flow field estimated through a flow network [56], which projects a location $\mathbf{p}$ in frame t to the location $\mathbf{p} + \delta\mathbf{p}$ in current frame i. The warping operation is implemented by the bilinear function applied on all the locations for each channel in the feature maps. The warping in certain channel is performed as:

$$\varphi^m_{t \to i}(\mathbf{p}) = \sum_{\mathbf{q}} K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p}) \varphi^m_t(\mathbf{q}) \tag{8}$$

where $\mathbf{p} = (p_x, p_y)$ means 2D locations, $\delta\mathbf{p} = Flow(I_t, I_i)(\mathbf{p})$ means flow in according positions, m indicates a channel in the feature maps $\varphi(\mathbf{x})$, $\mathbf{q} = (q_x, q_y)$ enumerates all spatial locations in the feature maps, and K means the bilinear interpolation kernel.

Since we adopt end-to-end training, the backpropagation of $\varphi_{t \to i}$ with respect to φ_t and flow $\delta\mathbf{p}$ (i.e $Flow(I_t, I_i)(\mathbf{p})$)is derived as:

$$\begin{aligned} \frac{\partial \varphi^m_{t \to i}(\mathbf{p})}{\partial \varphi^m_t(\mathbf{q})} =& K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p}) \\ \frac{\partial \varphi^m_{t \to i}(\mathbf{p})}{\partial Flow(I_t, I_i)(\mathbf{p})} =& \sum_{\mathbf{q}} \frac{\partial K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p})}{\partial \delta\mathbf{p}} \varphi^m_t(\mathbf{q}) \end{aligned} \tag{9}$$

Once the feature maps in previous frames are warped to current frames, they provides diverse information for same object instance, such as different viewpoints, deformation and varied illuminations. So appearance feature for tracked object can be enhanced by aggregating these feature maps. The aggregated at current frame is obtained as

$$\varphi_i = \sum_{t=i-T}^{i} w_{t \to i} \varphi_{t \to i} \tag{10}$$

where T is predefined intervals, $w_{t \to i}$ is adaptive weights at different spatial locations and feature channels. The adaptive weights are decided by proposed novel spatial-temporal attention mechanism which is described in detail in next subsection.

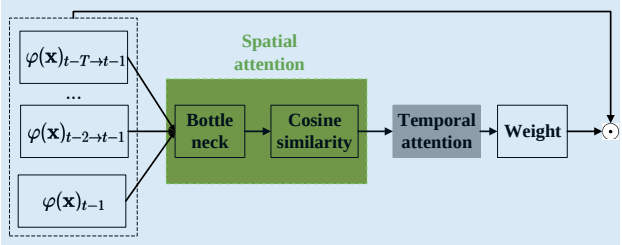

Figure 3: Spatial-temporal attention module.

3.4. Spatial-temporal attention

The adaptive weights indicate the importance of aggregated frames at each spatial location and temporal channels. For spatial location, we adopt cosine similarity metric to measure the similarity between the warped features and the features extracted from the current frame. For different channels, we further introduce temporal attention to adaptively re-calibrate temporal channels.

3.4.1 Spatial attention

Spatial attention indicates the different weights at different spatial locations. At first, a bottle neck sub-network projects the φ_i into a new embedding φ_i^e, then the cosine similarity metric is adopted to measure the similarity between the warped features and the features extracted from the current frame:

$$w_{t\to i}(\mathbf{p}) = SoftMax\left(\frac{\varphi_{t\to i}^e(\mathbf{p})\varphi_i^e(\mathbf{p})}{|\varphi_{t\to i}^e(\mathbf{p})|\,|\varphi_i^e(\mathbf{p})|}\right) \tag{11}$$

where $SoftMax$ operation is applied at channels to normalize the weight $w_{t\to i}$ for each spatial location $\mathbf{p}$ over the nearby frames. Intuitively speaking, in spatial attention, if the warped features $\varphi_{t\to i}^e(\mathbf{p})$ is close to the features $\varphi_i^e(\mathbf{p})$, it is assigned to a larger weight. Otherwise, a smaller weight is assigned.

3.4.2 Temporal attention

The weight $w_{t\to i}$ obtained by spatial attention has largest value at each position in current frame because current frame is most similar with its own according to cosine measurement. We further propose temporal attention mechanism to solve this problem by adaptively re-calibrating temporal channel. The channel number of spatial attention out is equal to the aggregated frame numbers, and we expect to re-weight the channel importance by introducing temporal information. Specifically, the output of spatial attention module is first passed through a global pooling to produce a channel-wise descriptor. Then a two-layer fully connected (FC)is added, in which learned for each channel by a self-gating mechanism based on channel dependence. This is followed by re-weighting the original feature maps to generate the output of temporal attention module. It is noting

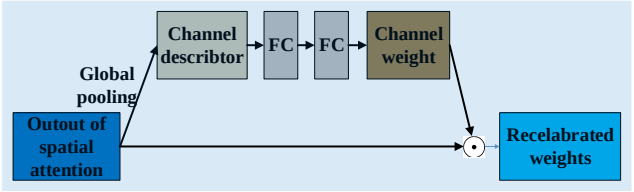

Figure 4: The temporal attention network architecture.

that our temporal attention mechanism is similar to SENet architecture in [41], while more parameters is adopted in FC layers.

4. Online Tracking

In this section, tracking network architecture is illustrated at first to describe our algorithm which is denoted as FlowTrack. Then we present the tracking process through the aspects of scale handing and model updating.

4.1. Overall tracking network architecture

After off-line training as described in Section 3, the learned network is used to perform online tracking by equation (4). At first, the images are passed through trained feature extraction network and Flow network. Then the feature maps in previous frames are warped to the current one according to flow information. Warped feature maps as well as the current frame's are embedded and then weighted using spatial-temporal attention. The estimate of the current target state is obtained by finding the maximum response in the score map. The CF layer in Figure 2 is replaced by standard CF tracking module.

4.2. Model updating

Most of tracking approaches update their model in each frame or at a fixed interval [9, 17, 19, 22, 23]. However, this strategy may introduce false background information when the tracking is inaccurate, target is occluded or out of view. In this paper, model update is decided by evaluating the tracking results. Specifically, we consider the maximum value in the response map and the distribution of other response value simultaneously [61].

Ideal response map should have only one peak value in actual target position and the other values are small. On the contrary, the response will fluctuate intensely and include more peak values. A criterion called peak-versus-noise ratio (*PNR*) is introduced to reveal the distribution of response map. The *PNR* is defined as

$$PNR = \frac{R_{max} - R_{min}}{mean(R \backslash R_{max})} \tag{12}$$

where

$$R_{max} = \max R(z) \tag{13}$$

and R_{min} is corresponding minimum value of response map. Denominator in equation (12) represents mean value

of response map except maximum value and is used to measure the noise approximately. The *PNR* criterion becomes larger when response map has fewer noise and sharper peak. Otherwise, the *PNR* criterion will fall into a smaller value. *PNR* criterion can significantly avoids unwanted updating in unreliable frames. We save the *PNR* and R_{max} and calculate their historical average values as threshold:

$$\begin{cases} PNR_{threshold} = \frac{\sum_{t=1}^{T} PNR_t}{T} \\ R_{threshold} = \frac{\sum_{t=1}^{T} R_{max}^t}{T} \end{cases} \tag{14}$$

When *PNR* and R_{max} at current frame exceed corresponding threshold, model update is performed as

$$\mathbf{f}^l = \mathcal{F}^{-1}\left(\frac{\sum_{t=1}^{p} \alpha_t \hat{\varphi}^l(\mathbf{x}_t) \circ \hat{\mathbf{y}}_t^*}{\sum_{t=1}^{p} \alpha_t (\sum_{k=1}^{D} \hat{\varphi}^k(\mathbf{x}_t) \circ (\hat{\varphi}^k(\mathbf{x}_t))^* + \lambda)}\right) \tag{15}$$

where α_t represents the impact of sample $\mathbf{x}_t$.

5. Experiments

Experiments are performed on four challenging tracking datasets: OTB2013 with 50 videos, OTB2015 with 100 videos, VOT2015 and VOT2016 with 60 videos . All the tracking results use the reported results to ensure a fair comparison.

5.1. Implement details

We adopt three convolution layers ($3 \times 3 \times 128, 3 \times 3 \times 128, 3 \times 3 \times 96$) in feature CNN and FlowNet follows the implementation in [56]. Our training data comes from VID [40], containing the training and validation set. The frame number of aggregation is set to 5. In each frame, patch is cropped around ground truth with a 1.56 padding and resized into $128 * 128$. We apply stochastic gradient descent (SGD) with momentum of 0.9 to end-to-end train the network and set the weight decay λ to 0.005. The model is trained for 50 epochs with a learning rate of 10^{-5}. In online tracking, scale penalty and model updating rate is set to 0.9925 and 0.015, respectively. The proposed FlowTrack is implemented using MatConvNet [58] on a PC with an Intel i7 6700 CPU, 48 GB RAM, Nvidia GTX TITAN X GPU. Average speed of the tracker is 12 FPS and the code will be made publicly available.

5.2. Results on OTB

OTB2013 [8] contains 50 fully annotated sequences that are collected from commonly used tracking sequences. OTB2015 [4] is the extension of OTB2013 and contains 100 video sequences. Some new sequences are more difficult to track. The evaluation is based on two metrics: precision plot and success plot. The precision plot shows the percentage of frames that the tracking results are within certain distance determined by given threshold to the ground truth. The value when threshold is 20 pixels is always taken as the representative precision score. The success plot shows the ratios of successful frames when the threshold varies from 0 to 1, where a successful frame means its overlap is larger than this given threshold. The area under curve (AUC) of each success plot is used to rank the tracking algorithm.

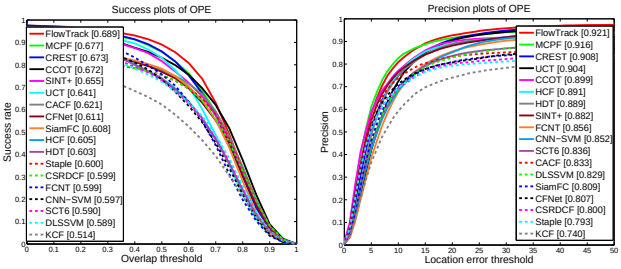

Figure 5: Performance on OTB2013.

5.2.1 Results of OTB2013

In this experiment, we compare our method against recent trackers that presented at top conferences and journals, including CREST (ICCV 2017) [62], MCPF (CVPR 2017) [45], UCT (ICCV 2017 Workshop) [61], CACF (CVPR 2017) [44], CFNet (CVPR 2017) [43], CSR-DCF (CVPR 2017) [59], CCOT (CVPR 2016) [22], SiamFC (ECCV 2016) [25], Staple (CVPR 2016) [18], SCT (CVPR 2016) [26], HDT (CVPR 2016) [20], DLSSVM (CVPR 2016) [30], SINT+ (CVPR 2016) [54], FCNT (ICCV 2015) [28], CNN-SVM (ICML 2015) [29],HCF (ICCV 2015) [19], KCF (T-PAMI 2015) [9]. The one-pass evaluation (OPE) is employed to compare these trackers.

Figure 5 illustrates the precision and success plots based on center location error and bounding box overlap ratio, respectively. It clearly illustrates that our algorithm, denoted by FlowTrack, outperforms the state-of-the-art trackers significantly in both measures. In the success plot, our approach obtain an AUC score of 0.689, significantly outperforms the winner of VOT2016 (CCOT) and another tracker using flow information (SINT+). The improvement ranges are 1.7% and 3.4%, respectively. In the precision plot, our approach obtains a score of 0.921, outperforms CCOT and SINT+ by 2.2% and 3.9%, respectively.

The top performance can be attributed to that our methods associating the rich flow information to improve the feature representation and the tracking accuracy. What is more, end-to-end training enables individual components in the tracking system is tightly coupled to work. By contrast, other trackers only consider appearance features, and hardly benefit from motion and interframe information. What is more, efficient updating and scale handling strategies ensure robustness of the tracker. It is worth noting that SINT+ adopts optical flow to filter out motion inconsistent candidates in Siamese tracking framework, while the optical flow is off-the-shelf and no end-to-end training is performed.

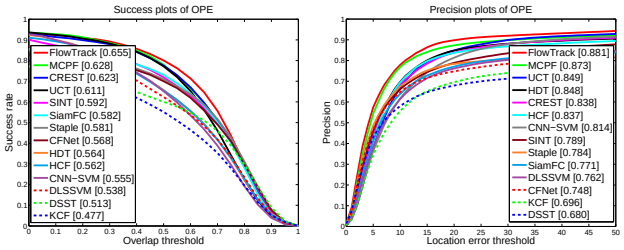

Figure 6: Attributes performance on OTB2015.

5.2.2 Results of OTB2015

In this experiment, we compare our method against recent trackers, including CREST (ICCV 2017) [62], CFNet (CVPR 2017) [43], MCPF (CVPR 2017) [45], UCT (ICCV 2017 Workshop) [61], DSST (T-PAMI 2017) [31], SiamFC (ECCV 2016) [25], Staple (CVPR 2016) [18], HDT (CVPR 2016) [20], SINT (CVPR 2016) [54], DLSSVM (CVPR 2016) [30], CNN-SVM (ICML 2015) [29], HCF (ICCV 2015) [19], KCF (T-PAMI 2015) [9]. The one-pass evaluation (OPE) is employed to compare these trackers.

Figure 6 illustrates the precision and success plots of the compared trackers, respectively. The proposed FlowTrack approach outperforms all the other trackers in terms of success and precision scores. Specifically, our method achieves a success score of 0.655, which outperforms the MCPF (0.628) and CREST (0.623) method with a large margin. For detailed performance analysis, we also report the results on various challenge attributes in OTB2015, such as occlusion, illumination variation, background clutter, etc. Figure 7 demonstrates that our tracker effectively handles these challenging situations while other trackers obtain lower scores. Comparisons of our approach with three state-of-the-art trackers in the challenging scenario is shown in Figure 1.

5.3. Results on VOT

The Visual Object Tracking (VOT) challenges are well-known competitions in tracking community, which have held several times from 2013 and their results will be reported at ICCV or ECCV. In this subsection, we compare our method, FlowTrack with entries in VOT2015 [5] and VOT2016 [34].

5.3.1 Results of VOT2015

VOT2015 [5] consists of 60 challenging videos that are automatically selected from a 356 sequences pool. The trackers in VOT2015 is evaluated by expected average overlap (EAO) measure, which is the inner product of the empirically estimating the average overlap and the typical-sequence-length distribution. The EAO measures the expected no-reset overlap of a tracker run on a short-term sequence. Besides, accuracy (mean overlap) and robustness (average number of failures) are also reported. In VOT2015 experiment, we present a state-of-the-art comparison to the participants in the challenge according to the latest VOT rules (see http://votchallenge.net). Figure 8 illustrates that our FlowTrack can rank *1st* in 61 trackers according to EAO criterion. It is worth noting that MDNet [21] is not compatible with the latest VOT rules because of OTB training data. In Table 1, we list the EAO, accuracy and failures number of FlowTrack and top 10 entries in VOT2015. FlowTrack rank *1st* according to all 3 criterions. The top performance can be attributed to the associating of flow information and end-to-end training framework.

Table 1: Comparision with top trackers in VOT2015. Red, green and blue fonts indicate *1st, 2nd, 3rd* performance, respectively.

Trackers	EAO	Accuracy	Failures
FlowTrack	0.3375	0.57	0.98
DeepSRDCF	0.3181	0.56	1.05
EBT	0.3130	0.47	1.02
srdcf	0.2877	0.56	1.24
LDP	0.2785	0.51	1.84
sPST	0.2767	0.55	1.48
scebt	0.2548	0.55	1.86
nsamf	0.2536	0.53	1.29
struck	0.2458	0.47	1.61
rajssc	0.2458	0.57	1.63
s3tracker	0.2420	0.52	1.77

5.3.2 Results of VOT2016

The datasets in VOT2016[34] are the same as VOT2015, but the ground truth bounding boxes have been re-annotated. As the same with VOT2015, in VOT 2016, each frame are annotated with six visual attributes, and the trackers are evaluated with accuracy, robustness and EAO. In experiment, we compare our method with participants in the challenge. As shown in Figure 9, FlowTrack can rank *5rd* in 70 trackers according to EAO criterion, which has a similar performance with CCOT and TCNN.

5.4. Ablation analyses

In this experiment, ablation analyses are performed to illustrate the effectiveness of proposed components. To verify the contribution of each component in our algorithm, we implement and evaluate four variations of our approach. At first, the baseline is implemented that no flow information is utilized(denoted by *no flow*). To verify the superiority of proposed flow aggregation and spatial-temporal attention strategy, we fuse the warped feature maps by decaying with time (denoted by *decay*). And the weight is obtained only by spatial attention, which is denoted as *no_ta*(*ta* means temporal attention). Lastly, model updating at each frame is adopted to compared with proposed *PNR* criterion (denoted as *no_PNR*). Analyses results include OTB2013 [8], OTB2015[4] VOT2015 [5]and VOT2016[6]. AUC means area under curve (AUC) of each success plot, and P20 represents precision score at 20 pixels.

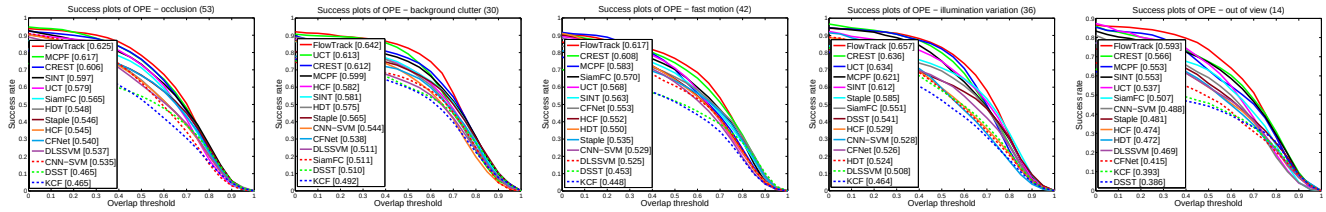

Figure 7: Attributes performance on OTB2015.

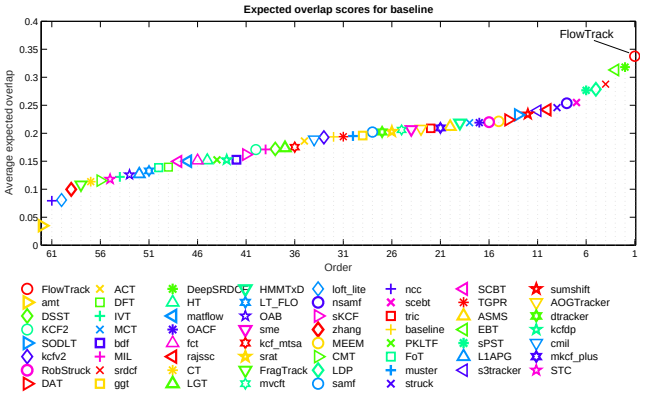

Figure 8: EAO ranking with trackers in VOT2015.

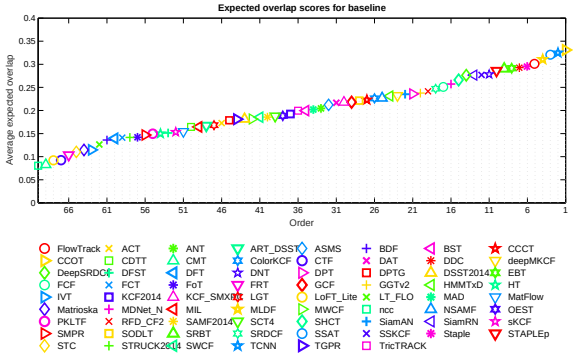

Figure 9: EAO ranking with trackers in VOT2016.

As shown in Table 2, the performances of all the variations are not as good as our full algorithm (denoted by *FlowTr*) and each component in our tracking algorithm is helpful to improve performance. Specifically, the associating and assembling of the flow information gains the performance with more than 6% in all evaluation criterions. The superiority of proposed flow aggregation is verified by gaining the EAO in 2015 and 2016 by near 7% and 5%, respectively. Temporal attention and model updating within *PNR* further improve the tracking performance.

5.5. Qualitative Results

To visualize the superiority of our framework on tracking performance, we show examples of FlowTrack results compared to recent trackers on challenging sample videos. As shown in Figure 1, the target in sequence *singer2* undergoes severe deformation. CCOT and CFNet lose the target from *#54* and CREST can not fit the scale changes. In contrast, the proposed FlowTrack results in successful tracking in this sequence because feature representation is enhanced using flow information. *skating1* is a sequences with attributes of illumination and pose variations, and proposed method can handle these challenges while CCOT drift to background. In sequence *carscale*, Only FlowTrack can handle the scale challenges in *#197* and *#252*. In background clutter of sequence *bolt2*, FlowTrack tracks the target successfully while compared approaches drifts to distractors.

Table 2: Performance on benchmarks of *FlowTrack* and its variations

	OTB2013 AUC	OTB2013 P20	OTB2015 AUC	OTB2015 P20	VOT2015 EAO	VOT2016 EAO
no flow	0.625	0.846	0.578	0.792	0.2637	0.2404
decay	0.637	0.868	0.586	0.793	0.2584	0.2516
no_PNR	0.670	0.898	0.621	0.866	0.3283	0.2825
no_ta	0.667	0.874	0.642	0.865	0.3109	0.2712
FlowTr	0.689	0.921	0.655	0.881	0.3375	0.3024

6. Conclusions

In this work, we propose an end-to-end framework for tracking which associates and assembles the rich flow information in consecutive frames. Specifically, the frames in certain intervals are warped to current frame using flow information and then they are aggregated for consequent correlation filter tracking. Meanwhile, a novel spatial-temporal attention mechanism is developed to adaptively weigh warped feature maps as well as current feature maps. All the modules, including optical flow estimation, feature extraction and aggregation, correlation filter tracking are trained end-to-end. In experiments, our method achieves state-of-the-art results on OTB2013, OTB2015, VOT2015 and VOT2016.

References

[1] Renoust B, Le D D, Satoh S. Visual analytics of political networks from face-tracking of news video. IEEE Transactions on Multimedia, 2016, 18(11): 2184-2195. 1

[2] Lee K H, Hwang J N. On-road pedestrian tracking across multiple driving recorders. IEEE Transactions on Multimedia, 2015, 17(9): 1429-1438. 1

[3] Iqbal U, Milan A, Gall J. Pose-Track: Joint Multi-Person Pose Estimation and Tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1

[4] Wu Y, Lim J, Yang M H. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848. 1, 6, 7

[5] Kristan M, Matas J, Leonardis A, et al. The visual object tracking vot2015 challenge results. Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015: 1-23. 7

[6] Kristan M, Leonardis A, Jiri Matas, et al. The visual object tracking vot2016 challenge results. Proceedings of the ECCV Workshops. 2016: 1-45. 7

[7] Smeulders A W M, Chu D M, Cucchiara R, et al. Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(7): 1442-1468. 1

[8] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013: 2411-2418. 6, 7

[9] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596. 1, 2, 5, 6, 7

[10] Danelljan M, Hager G, Shahbaz Khan F, et al. Learning spatially regularized correlation filters for visual tracking. Proceedings of the IEEE International Conference on Computer Vision. 2015: 4310-4318. 1, 2

[11] Ma C, Yang X, Zhang C, et al. Long-term correlation tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5388-5396. 1, 2

[12] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012: 1097-1105. 1, 2

[13] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778. 1, 2

[14] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems. 2015: 91-99. 1, 2

[15] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440. 1

[16] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In IEEE Conference on Computer Vision and Pattern Recognition, 2010. 2

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In ECCV, 2012. 1, 2, 5

[18] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary learners for real-time tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1401-1409. 2, 6, 7

[19] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3074-3082. 1, 2, 3, 5, 6, 7

[20] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang. Hedged deep tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 4303-4311. 1, 2, 3, 6, 7

[21] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4293-4302. 3, 7

[22] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking. European Conference on Computer Vision. Springer International Publishing, 2016: 472-488. 3, 5, 6

[23] Danelljan M, Bhat G, Khan F S, et al. ECO: Efficient Convolution Operators for Tracking. in CVPR 2017. 3, 5

[24] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking. Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015: 58-66. 1, 2, 3

[25] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision, 2016: 850-865. 3, 6, 7

[26] Held D, Thrun S, Savarese S. Learning to track at 100 fps with deep regression networks. European Conference on Computer Vision, 2016: 749-765. 3, 6

[27] Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration. European Conference on Computer Vision, 2014: 254-265. 1, 2

[28] Wang L, Ouyang W, Wang X, et al. Visual tracking with fully convolutional networks. Proceedings of the IEEE International Conference on Computer Vision. 2015: 3119-3127. 6

[29] Hong S, You T, Kwak S, et al. Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network. ICML. 2015: 597-606. 6, 7

[30] Ning J, Yang J, Jiang S, et al. Object tracking via dual linear structured SVM and explicit feature map. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4266-4274. 6, 7

[31] Danelljan M, Hager G, Khan F S, et al. Discriminative Scale Space Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 2, 7

[32] Kiani Galoogahi H, Sim T, Lucey S. Correlation filters with limited boundaries. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015:4630-4638.

[33] Wang Q, Gao J, Xing J, et al. DCFNet: Discriminant Correlation Filters Network for Visual Tracking[J]. arXiv preprint arXiv:1704.04057, 2017. 3, 4

[34] Kristan M, Leonardis A, Jiri Matas, et al. The visual object tracking vot2016 challenge results[C] Proceedings of the ECCV Workshops. 2016: 1-45. 7

[35] Brox T, Bruhn A, Papenberg N, et al. High accuracy optical flow estimation based on a theory for warping[J]. Computer Vision-ECCV 2004, 2004: 25-36. 2

[36] Horn B K P, Schunck B G. Determining optical flow[J]. Artificial intelligence, 1981, 17(1-3): 185-203. 2

[37] Revaud J, Weinzaepfel P, Harchaoui Z, et al. Epicflow: Edge-preserving interpolation of correspondences for optical flow, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1164-1172. 2

[38] Weinzaepfel P, Revaud J, Harchaoui Z, et al. DeepFlow: Large displacement optical flow with deep matching, Proceedings of the IEEE International Conference on Computer Vision. 2013: 1385-1392. 2

[39] Danelljan M, Shahbaz Khan F, Felsberg M, et al. Adaptive color attributes for real-time visual tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1090-1097. 2

[40] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211-252.

[41] Hu J, Shen L, Sun G. Squeeze-and-Excitation Networks. arXiv preprint arXiv:1709.01507, 2017.

[42] Danelljan M, Bhat G, Khan F S, et al. ECO: Efficient Convolution Operators for Tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 6

[43] Valmadre J, Bertinetto L, Henriques J F, et al. End-to-end representation learning for Correlation Filter based tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 5

[44] Mueller M, Smith N, Ghanem B. Context-Aware Correlation Filter Tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[45] Zhang T, Xu C, Yang M H. Multi-task Correlation Particle Filter for Robust Object Tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1, 2, 3, 6, 7

[46] Hong Z, Chen Z, Wang C, et al. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 749-758. 1, 2, 6

[47] Choi J, Jin Chang H, Jeong J, et al. Visual tracking using attention-modulated disintegration and integration[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4321-4330. 2, 6, 7

[48] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 2544-2550. 1, 2

[49] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In ICCV, 2015. 2

[50] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. ICLR 2016 Workshop.

[51] W. Zhang, P. Srinivasan, and J. Shi. Discriminative image warping with attribute flow. In CVPR, 2011. 2, 3

[52] Zhu X, Xiong Y, Dai J, et al. Deep feature flow for video recognition. CVPR2017, 2017. 2, 3

[53] Zhu X, Wang Y, Dai J, et al. Flow-Guided Feature Aggregation for Video Object Detection. ICCV2017, 2017. 2, 3

[54] Tao R, Gavves E, Smeulders A W M. Siamese instance search for tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1420-1429. 3

[55] Gladh S, Danelljan M, Khan F S, et al. Deep motion features for visual tracking[C]//Pattern Recognition (ICPR), 2016 23rd International Conference on. IEEE, 2016: 1243-1248. 3

[56] Dosovitskiy A, Fischer P, Ilg E, et al. Flownet: Learning optical flow with convolutional networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 2758-2766. 2, 3, 6, 7

[57] Ilg E, Mayer N, Saikia T, et al. Flownet 2.0: Evolution of optical flow estimation with deep networks. CVPR2017. 2, 3

[58] Vedaldi A, Lenc K. Matconvnet: Convolutional neural networks for matlab[C]//Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015: 689-692. 2, 4, 6

[59] Alan Lukezic, et al. Discriminative Correlation Filter with Channel and Spatial Reliability. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2

[60] Kiani Galoogahi H, Fagg A, Lucey S. Learning Background-Aware Correlation Filters for Visual Tracking. IEEE International Conference on Computer Vision, 2017. 6

[61] Zheng Zhu, Guan Huang, Wei Zou, Dalong Du and Chang Huang. IEEE International Conference on Computer Vision Workshops, 2017. 2, 6

[62] Song Y, Ma C, Gong L, et al. CREST: Convolutional Residual Learning for Visual Tracking. IEEE International Conference on Computer Vision, 2017. 2