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

Zheng Zhu[1,2], Wei Wu[3], Wei Zou[1,2,4], Junjie Yan[3]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]SenseTime Group Limited, Beijing, China
[4]TianJin Intelligent Tech.Institute of CASIA Co.,Ltd, Tianjin, China
{zhuzheng2014,wei.zou}@ia.ac.cn {wuwei,yanjunjie}@sensetime.com

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 inter-frame information. The lack of temporal information degrades the tracking performance during challenges such as partial occlusion and deformation. In this paper, we propose the FlowTrack, which focuses on making use of the rich flow information in consecutive frames to improve the feature representation and the tracking accuracy. The FlowTrack formulates individual components, including optical flow estimation, feature extraction, aggregation and correlation filters tracking as special layers in network. To the best of our knowledge, this is the first work to jointly train flow and tracking task in deep learning framework. Then the historical feature maps at predefined intervals are warped and aggregated with current ones by the guiding of flow. For adaptive aggregation, we propose a novel spatial-temporal attention mechanism. In experiments, the proposed method achieves leading performance on OTB2013, OTB2015, VOT2015 and VOT2016.

1. Introduction

Visual object tracking, which tracks a specified target in a changing video sequence automatically, is a fundamental problem in many computer vision topics such as visual analysis, automatic driving, pose estimation. A core problem of tracking is how to detect and locate the object accurately in changing scenarios with occlusions, shape deformation, illumination variations [42, 20].

Recently, significant attention has been paid to discriminative correlation filters (DCF) based methods for visual tracking such as KCF [14], SAMF [22], LCT [26], MUSTer [17], SRDCF [7] and CACF [27]. Most of these methods use handcrafted features, which hinder their accuracy and robustness. Inspired by the success of convolution neural networks (CNN) in object recognition, the visual tracking community has been focused on the deep trackers that exploit the strength of CNN in recent years. Representative deep trackers include DeepSRDCF [5], HCF [25], SiamFC [2] and CFNet [37]. However, most existing trackers only consider appearance features of current frame, and can hardly benefit from motion and inter-frame information. The lack of temporal information degrades the tracking performance during challenges such as partial occlusion and deformation. Although some trackers utilize optical flow to upgrade performance [36, 11], the flow feature is off-the-shelf and not trained end-to-end. These methods do not take full advantage of flow information.

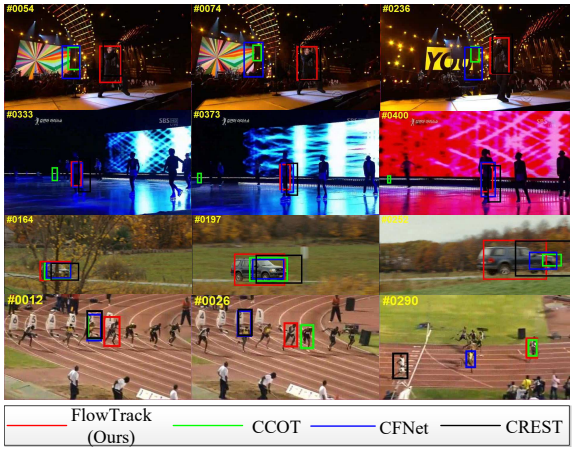

Figure 1: Tracking results comparison of our approach with three state-of-the-art trackers in the challenging scenarios. Best viewed on color display.

In this paper, we develop an end-to-end flow correlation tracking framework (FlowTrack) to utilize both the flow information and appearance features, which improves the feature representation and tracking accuracy. Specifically, we

formulate the optical flow estimation, feature extraction, aggregation and correlation filter tracking as special layers in network, which enables end-to-end learning. Then the previous frames are warped to specified frame guided by flow information, and they are aggregated for consequent correlation filter tracking. For adaptive aggregation, a novel spatial-temporal attention mechanism is developed. In spatial attention, feature maps are weighted in planar position using spatial similarities. Channels of feature maps are then re-weighted to take temporal attention into account.

Features from different frames provide 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 features. 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 because of misalignment. This suggests that it is critical to model the motion during learning. To this end, we propose to end-to-end train the flow estimation and adaptive feature aggregation using large-scale tracking dataset. Figure 1 shows four challenging benchmark sequences which undergo illumination variation, viewpoint changes and deformation. The FlowTrack can handle these challenges due to the aggregation of diverse feature maps. In experiments, we achieve leading performance on four challenging tracking benchmarks.

The contributions of this paper can be summarized in three folds. 1) We develop an end-to-end flow correlation tracking framework 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 deep learning framework. 2) A novel spatial-temporal attention mechanism is proposed, which can adaptively aggregate the warped and current feature maps. 3) Experiments on OTB2013, OTB2015, VOT2015 and VOT2016 shows that the proposed method performs favorably against existing state-of-the-art methods.

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 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 trackers

In recent tracking community, significant attention has been paid to discriminative correlation filters (DCF) based methods [3, 15, 14, 22, 6, 9, 1, 26, 17, 7, 27] because of their efficiency and expansibility. MOSSE [3], CSK [15] and KCF [14] are conventional DCF trackers. Many improvements for DCF tracking approaches have been proposed, such as SAMF [22] and fDSST [6] for scale changes, CN [9] and Staple [1] taking color information into account, LCT [26] and MUSTer [17] for long-term tracking, SRDCF [7] and CACF [27]to mitigate boundary effects. Most of these methods use handcrafted features, which hinder their accuracy and robustness.

Inspired by the success of CNN in object classification [21, 12], detection [33] and segmentation [23] tasks, 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 [25] and HDT [32], 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 [5] exploits shallow CNN features in a spatially regularized DCF framework. In above mentioned methods, the chosen CNN features are always pre-trained in different tasks and individual components in tracking systems are learned separately. So the achieved tracking results may be suboptimal. It is worth noting that CFNet [37] and DCFNet [40] interpret the correlation filters as a differentiable layer in a Siamese tracking framework, thus achieving an end-to-end representation learning. The main drawback is their unsatisfying performance.

2.2. CNN-based trackers

Except 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 [2] 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 [13], the motion between successive frames is predicted using a deep regression network. MDNet [28] trains a small-scale network by multi-domain methods, thus separating domain independent information from domain-specific layers. CCOT [8] employs the implicit interpolation method to solve the learning problem in the continuous spatial domain. CREST [35] treats tracking process as convolution and applies residual learning to take appearance changes into account. Similarly, UCT [48] treats feature extractor and tracking process both as convolution operation and trains them jointly, enabling learned CNN features tightly coupled to tracking process. All these trackers only consider appearance features in current frame and can hardly benefit from motion and inter-frame information. In this paper, we

make full use of these information by aggregating flow and correlation tracking in an end-to-end framework.

2.3. Optical flow for visual recognition

Flow information has been exploited to be helpful in computer vision tasks. In pose estimation [31], optical flow is used to align heatmap predictions from neighbouring frames. [30] applies flow to the current frame to predict next frame. In [45], flow is used to explicitly model how image attributes vary with its deformation. DFF [47] and FGFA [46] 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 are 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 [36, 11], while the flow feature is off-the-shelf and not trained end-to-end. In this paper, we formulate the optical flow estimation in an end-to-end tracking framework and model the motion during learning.

3. End-to-end 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 aggregated frames at each spatial location and temporal channels, a novel spatial-temporal attention mechanism is designed. At last, online tracking is described consisting of model updating and scales.

3.1. Training network architecture

The overall training framework of our tracker consists of FeatureNet (feature extraction sub-network), FlowNet [10], warping module, spatial-temporal attention module and CF tracking layer. As shown in Figure 2, overall training architecture adopts Siamese network consisting of historical and current branches. In historical branch, appearance features and flow information are extracted by the FeatureNet and FlowNet at first. Then previous frames at predefined intervals (5 frames in experiments, $T = 6$) is warped to $t-1$ frame guided by flow information. Meanwhile, a spatial-temporal attention module is designed to weight the warped feature maps. In another branch, the feature maps of current frame is extracted by FeatureNet. Finally, both two branches are fed into subsequent correlation filters layer for training. All the modules are differentiable and trained end-to-end.

3.2. Correlation filters layer

Discriminative correlation filters (DCF) with deep convolutional features have shown favorable performance in recent benchmarks [25, 32, 5]. Nonetheless, the chosen CNN features are always pre-trained in different tasks and individual components in tracking systems are learned separately, thus the achieved tracking results may be suboptimal. Recently, CFNet [37] and DCFNet [40] interpret the correlation filters as a differentiable layer in Siamese framework, thus performing 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 the FeatureNet 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 filters 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 filters can be trained by minimizing error which is obtained between the response $R(\mathbf{x}_k)$ on sample $\mathbf{x}_k$ and the corresponding Gaussian label $\mathbf{y}_k$:

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

The second term in (2) is a regularization with a weight parameter λ. The solution can be gained as [6]:

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

where the hat symbol represents the discrete Fourier transform $\mathcal{F}$ of according variables, * represents the complex conjugate of according variables, D is the channel numbers, 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 $\varphi(\mathbf{z})$ denotes the feature maps 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 correlation filters layer. Given the feature maps of search patch $\varphi(\mathbf{z})$, the loss

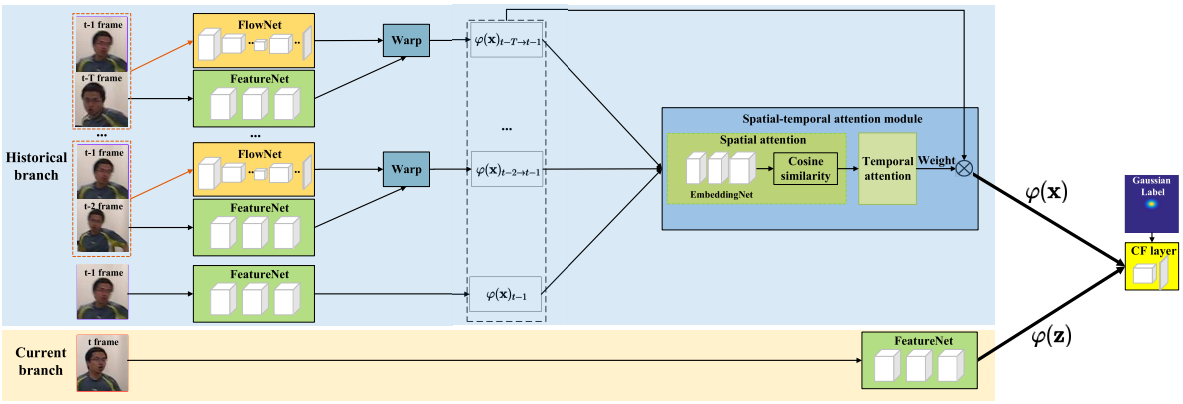

Figure 2: The overall training network. The network adopts Siamese architecture consisting of historical and current branches. The dashed boxes in left part represent concatenating two input frames for FlowNet, and the feature maps in dashed box (middle part) are weighted by output of spatial-temporal attention module. Best viewed on color display.

function is formulated as:

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

where $\tilde{R}$ is desired response, and it is a gaussian distribution centered at the real target location. $\boldsymbol{\theta}$ refers to the parameters of the whole network. The back-propagation of loss with respect to $\varphi(\mathbf{x})$ and $\varphi(\mathbf{z})$ are formulated as [40]:

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

Once the back-propagation is derived, the correlation filters can be formulated as a layer in network, which is called CF 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 specified frame according to the flow:

$$\varphi_{i\to t-1} = \mathcal{W}(\varphi_i, Flow(I_i, I_{t-1})) \tag{7}$$

where $\varphi_{i\to t-1}$ denotes the feature maps warped from previous frame i to specified $t-1$ frame. $Flow(I_i, I_{t-1})$ is the flow field estimated through a flow network [10], which projects a location $\mathbf{p}$ in frame i to the location $\mathbf{p} + \delta\mathbf{p}$ in specified frame $t-1$. 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:

$$\varphi^m_{i\to t-1}(\mathbf{p}) = \sum_{\mathbf{q}} K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p})\varphi^m_i(\mathbf{q}) \tag{8}$$

where $\mathbf{p} = (p_x, p_y)$ means 2D locations, and $\delta\mathbf{p} = Flow(I_i, I_{t-1})(\mathbf{p})$ represents 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 indicates the bilinear interpolation kernel.

Since we adopt end-to-end training, the back-propagation of $\varphi_{i\to t-1}$ with respect to φ_i and flow $\delta\mathbf{p}$ (*i.e.* $Flow(I_i, I_{t-1})(\mathbf{p})$) is derived as:

$$\begin{aligned} \frac{\partial \varphi^m_{i\to t-1}(\mathbf{p})}{\partial \varphi^m_i(\mathbf{q})} &= K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p}) \\ \frac{\partial \varphi^m_{i\to t-1}(\mathbf{p})}{\partial Flow(I_i, I_{t-1})(\mathbf{p})} &= \sum_{\mathbf{q}} \frac{\partial K(\mathbf{q}, \mathbf{p} + \delta\mathbf{p})}{\partial \delta\mathbf{p}} \varphi^m_i(\mathbf{q}) \end{aligned} \tag{9}$$

Once the feature maps in previous frames are warped to specified frame, they provide 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 aggregation results are obtained as:

$$\varphi(\mathbf{x}) = \overline{\varphi}_{t-1} = \sum_{i=t-T}^{t-1} w_{i\to t-1}\varphi_{i\to t-1} \tag{10}$$

where T is predefined intervals, $w_{i\to t-1}$ 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.

3.4. Spatial-temporal attention

The adaptive weights indicate the importance of aggregated frames at each spatial location and temporal channel-

s. For spatial location, we adopt cosine similarity metric to measure the similarity between the warped features and the features extracted from the specified $t-1$ frame. For different channels, we further introduce temporal attention to adaptively re-calibrate temporal channels [18].

3.4.1 Spatial attention

Spatial attention indicates the different weights at different spatial locations. At first, a bottleneck sub-network projects the φ into a new embedding φ^e, then the cosine similarity metric is adopted to measure the similarity between the warped features and the features extracted from the specified $t-1$ frame:

$$w_{i\to t-1}(\mathbf{p}) = SoftMax\left(\frac{\varphi^e_{i\to t-1}(\mathbf{p})\varphi^e_{t-1}(\mathbf{p})}{\left|\varphi^e_{i\to t-1}(\mathbf{p})\right|\left|\varphi^e_{t-1}(\mathbf{p})\right|}\right) \tag{11}$$

where $SoftMax$ operation is applied at channels to normalize the weight $w_{i\to t-1}$ for each spatial location $\mathbf{p}$ over the nearby frames. Intuitively speaking, in spatial attention, if the warped features $\varphi^e_{i\to t-1}(\mathbf{p})$ is close to the features $\varphi^e_{t-1}(\mathbf{p})$, it is assigned with a larger weight. Otherwise, a smaller weight is assigned.

3.4.2 Temporal attention

The weight $w_{i\to t-1}$ obtained by spatial attention has largest value at each position in $t-1$ frame because $t-1$ 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 as shown in Figure 3. The channel number of spatial attention out is equal to the aggregated frame numbers T, and we expect to re-weight the channel importance by introducing temporal information.

Specifically, the output of spatial attention module is firstly passed through a global pooling layer to produce a channel-wise descriptor. Then three fully connected (FC) layers are 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.

The weights in temporal frames (channels) are visualized to illustrate the results of our temporal attention. In Figure 4, the first and second row indicate the normal and challenging scenarios, respectively. As shown in top left corner in each frames, the weights are approximately equal in normal scenarios. In challenging scenarios, the weights are smaller in low quality frames while larger in high quality frames, which shows re-calibration role of the temporal attention module.

3.5. Online Tracking

In this subsection, tracking network architecture is described at first which is denoted as FlowTrack. Then we present the tracking process through the aspects of scale handing and model updating.

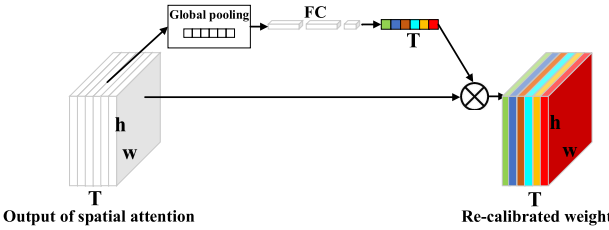

Figure 3: The temporal attention sub-network architecture. Channels with different colors are re-calibrated by different weights. Best viewed on color display.

Figure 4: The visualization of weights in temporal frames (channels). The first and second row show normal and challenging scenarios, respectively. The number in top left corner indicates learned temporal weights. Best viewed on color display.

Tracking network architecture After off-line training as described above, the learned network is used to perform online tracking by equation (4). At first, the images are passed through trained FeatureNet and FlowNet. 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 estimation of the current target state is obtained by finding the maximum response in the score map.

Model updating Most of tracking approaches update their model in each frame or at a fixed interval [15, 14, 25, 8]. However, this strategy may introduce false background information when the tracking is inaccurate, target is occluded or out of view. In this paper, model updating is performed when criterions peak-versus-noise ratio (*PNR*) and maximum value of response map are satisfied at the same time. Readers are referred to [48] for details. Only CF tracking module is updated as:

$$\mathbf{f}^l = \mathcal{F}^{-1}\left(\frac{\sum_{t=1}^{p}\alpha_t\hat{\varphi}^l(\mathbf{x}_t)\odot\hat{\mathbf{y}}_t{}^*}{\sum_{t=1}^{p}\alpha_t(\sum_{k=1}^{D}\hat{\varphi}^k(\mathbf{x}_t)\odot(\hat{\varphi}^k(\mathbf{x}_t))^*+\lambda)}\right) \tag{12}$$

where α_t represents the impact of sample $\mathbf{x}_t$, and p equals to the frame index.

Scales To handle the scale change, we follow the approach in [43] and use patch pyramid with the scale factors $\{a^s \mid s = \lfloor -\frac{S-1}{2} \rfloor, \lfloor -\frac{S-3}{2} \rfloor, ..., 0, ..., \lfloor \frac{S-1}{2} \rfloor\}$.

4. 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.

4.1. Implementation details

We adopt three convolution layers ($3 \times 3 \times 128, 3 \times 3 \times 128, 3 \times 3 \times 96$) in FeatureNet, and FlowNet follows the implementation in [10]. Embedding sub-network in spatial attention consists of three convolution layers ($1 \times 1 \times 64, 3 \times 3 \times 64, 1 \times 1 \times 256$) which are randomly initialized. Fully connected (FC) layers in temporal attention is set to $1 \times 1 \times 128, 1 \times 1 \times 128, 1 \times 1 \times 6$. First two and last FC layer are followed by ReLU and Sigmoid, respectively. Our training data comes from VID [34], containing the training and validation set. The frame number of aggregation is set to 5 (T in Figure 2 is set to 6). 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 step a and number S is set to 1.025 and 5, scale penalty and model updating rate is set to 0.9925 and 0.015. The proposed FlowTrack is implemented using MatConvNet [38] 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 experimental results can be found in https://github.com/zhengzhugithub/FlowTrack.

4.2. Results on OTB

OTB2013 [41] contains 50 fully annotated sequences that are collected from commonly used tracking sequences. OTB2015 [42] 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.

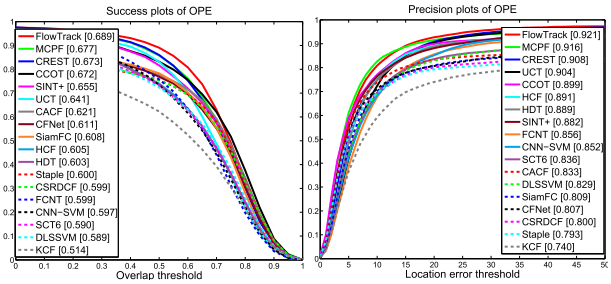

Figure 5: Precision and success plots on OTB2013. The numbers in the legend indicate the representative precisions at 20 pixels for precision plots, and the area-under-curve scores for success plots. Best viewed on color display.

4.2.1 Results of OTB2013

In this experiment, we compare our method against recent trackers that presented at top conferences and journals, including CREST (ICCV 2017) [35], MCPF (CVPR 2017) [44], UCT (ICCV 2017 Workshop) [48], CACF (CVPR 2017) [27], CFNet (CVPR 2017) [37], CSR-DCF (CVPR 2017) [24], CCOT (CVPR 2016) [8], SiamFC (ECCV 2016) [2], Staple (CVPR 2016) [1], SCT (CVPR 2016) [4], HDT (CVPR 2016) [32], DLSSVM (CVPR 2016) [29], SINT+ (CVPR 2016) [36], FCNT (ICCV 2015) [39], CNN-SVM (ICML 2015) [16], HCF (ICCV 2015) [25], KCF (T-PAMI 2015) [14]. 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 method makes use of 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 are tightly coupled to work. By contrast, other trackers only consider appearance features, and hardly benefit from motion and inter-frame 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.

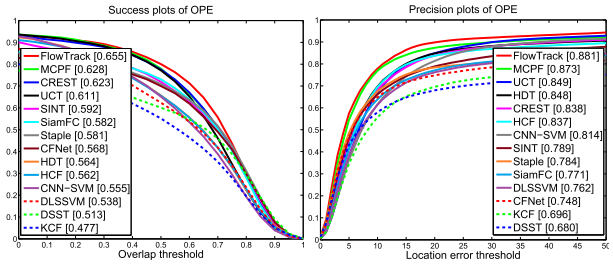

Figure 6: Precision and success plots on OTB2015. The numbers in the legend indicate the representative precisions at 20 pixels for precision plots, and the area-under-curve scores for success plots. Best viewed on color display.

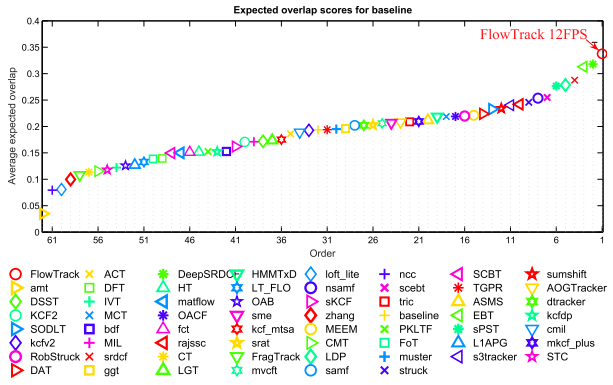

Figure 7: EAO ranking with trackers in VOT2015. The better trackers are located at the right. Best viewed on color display.

4.2.2 Results of OTB2015

In this experiment, we compare our method against recent trackers, including CREST (ICCV 2017) [35], CFNet (CVPR 2017) [37], MCPF (CVPR 2017) [44], UCT (ICCV 2017 Workshop) [48], DSST (T-PAMI 2017) [6], SiamFC (ECCV 2016) [2], Staple (CVPR 2016) [1], HDT (CVPR 2016) [32], SINT (CVPR 2016) [36], DLSSVM (CVPR 2016) [29], CNN-SVM (ICML 2015) [16], HCF (ICCV 2015) [25], KCF (T-PAMI 2015) [14]. 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.

4.3. Results on VOT

The Visual Object Tracking (VOT) challenges are well-known competitions in tracking community, which have held several times since 2013 and their results will be reported at ICCV or ECCV. In this subsection, we compare our method, FlowTrack, with entries in VOT2015 [20] and VOT2016 [19].

Table 1: Comparisons with top trackers in VOT2015. Red, green and blue fonts indicate *1st, 2nd, 3rd* performance, respectively. Best viewed on color display.

Trackers	EAO	Accuracy	Failures
FlowTrack	0.3405	0.57	0.95
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

4.3.1 Results of VOT2015

VOT2015 [20] 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 with the participants in the challenge according to the latest VOT rules (see http://votchallenge.net). Figure 7 illustrates that our FlowTrack can rank *1st* in 61 trackers according to EAO criterion. It is worth noting that MDNet [28] is not compatible with the latest VOT rules because of OTB training data. In Table 1, we list the EAO, accuracy and failures of FlowTrack and top 10 entries in VOT2015. FlowTrack ranks *1st* according to all 3 criterions. The top performance can be attributed to the associating of flow information and end-to-end training framework.

4.3.2 Results of VOT2016

The datasets in VOT2016[19] are the same as VOT2015, but the ground truth has been re-annotated. VOT2016 also adopts EAO, accuracy and robustness for evaluations.

In experiment, we compare our method with participants in challenges. Figure 8 illustrates that our FlowTrack can rank *1st* in 70 trackers according to EAO criterion. It is worth noting that our method can operate at 12 FPS, which is 40 times faster than CCOT [8] (0.3 FPS). For detailed performance analysis, we further list accuracy and robustness of representative trackers in VOT2016. As shown in Table 2, the accuracy and robustness of proposed FlowTrack can rank *1st* and *2nd*, respectively.

4.4. Ablation analyses

In this experiment, ablation analyses are performed to illustrate the effectiveness of proposed components. To verify the contributions of each component in our algorithm, we implement and evaluate four variations of our approach. At first, the baseline is implemented that no flow informa-

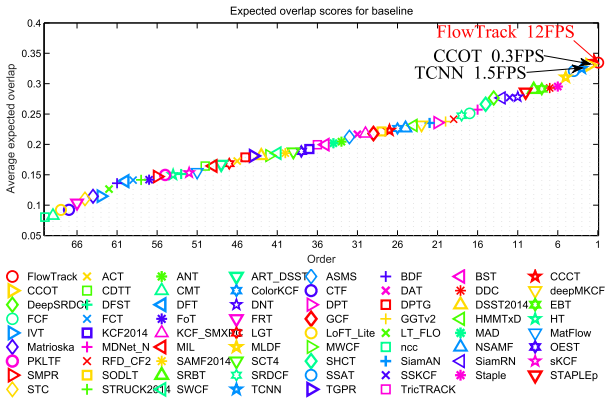

Figure 8: EAO ranking with trackers in VOT2016. The better trackers are located at the right. Best viewed on color display.

Table 2: Comparisons with top trackers in VOT2016. Red, green and blue fonts indicate *1st, 2nd, 3rd* performance, respectively. Best viewed on color display.

Trackers	EAO	Accuracy	Robustness
FlowTrack	0.334	0.578	0.241
CCOT	0.331	0.539	0.238
TCNN	0.325	0.554	0.268
Staple	0.295	0.544	0.378
EBT	0.291	0.465	0.252
DNT	0.278	0.515	0.329
SiamFC	0.277	0.549	0.382
MDNet	0.257	0.541	0.337

tion is utilized(denoted by *no flow*). Then the FlowNet is fixed to compare with end-to-end training (denoted by *fix 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* (means no temporal attention). Analyses results include OTB2013 [41], OTB2015 [42], VOT2015 [20]and VOT2016[19]. AUC means area under curve (AUC) of each success plot, and P20 represents precision score at 20 pixels.

As shown in Table 3, 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, in terms of *no flow* and *FlowTr*, the associating and assembling of the flow information gains the performance with more than 6% in all evaluation criterions. In terms of *no flow*, *fix flow* and *FlowTr*, the performance of VOT even drops when FlowNet is added but fixed, which verifies the necessity of end-to-end training. Comparing *decay* with *FlowTr*, the superiority of proposed flow aggregation is verified by gaining the EAO in 2015 and 2016 by near 8%. Besides, temporal attention further improves the tracking performance.

4.5. Qualitative Results

To visualize the superiority of flow correlation filters framework, we show examples of FlowTrack results compared to recent trackers on challenging sample videos. As

Table 3: 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
fix flow	0.617	0.853	0.583	0.813	0.2542	0.2291
decay	0.637	0.868	0.586	0.793	0.2584	0.2516
no_ta	0.667	0.874	0.642	0.865	0.3109	0.2712
FlowTr	0.689	0.921	0.655	0.881	0.3405	0.3342

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 change. 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 drift to distracters.

5. Conclusions

In this work, we propose an end-to-end flow correlation tracking framework which makes use of the rich flow information in consecutive frames. Specifically, the frames in certain intervals are warped to specified frame using flow information and then they are aggregated for consequent correlation filter tracking. For adaptive aggregation, a novel spatial-temporal attention mechanism is developed. The effectiveness of our approach is validated in OTB and VOT datasets.

Acknowledgment

This work is supported in part by the National High Technology Research and Development Program of China under Grant No.2015AA042307, the National Natural Science Foundation of China under Grant No.61773374, and in part by Project of Development In Tianjin for Scientific Research Institutes Supported By Tianjin government under Grant No.16PTYJGX00050. This work is done when Zheng Zhu is an intern at SenseTime Group Limited.

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016. 2, 6, 7

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision Workshop*, pages 850–865, 2016. 1, 2, 6, 7

[3] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2544–2550, 2010. 2

[4] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi. Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4330, 2016. 6

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*, pages 621–629, 2015. 1, 2, 3

[6] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1561–1575, 2017. 2, 3, 7

[7] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015. 1, 2

[8] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proceedings of the European Conference on Computer Vision*, pages 472–488, 2016. 2, 5, 6, 7

[9] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2014. 2

[10] P. Fischer, A. Dosovitskiy, E. Ilg, P. Husser, C. Hazırba, V. Golkov, V. D. S. Patrick, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 3, 4, 6

[11] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg. Deep motion features for visual tracking. In *Proceedings of the International Conference on Pattern Recognition*, pages 1243–1248. IEEE, 2016. 1, 3

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[13] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *Proceedings of the European Conference on Computer Vision*, pages 749–765, 2016. 2

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

[15] J. F. Henriques, C. Rui, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the European Conference on Computer Vision*, pages 702–715, 2012. 2, 5

[16] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional neural network. In *Proceedings of the International Conference on Machine Learning*, pages 597–606, 2015. 6, 7

[17] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015. 1, 2

[18] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 5

[19] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. ehovin, T. Vojr, G. Hger, A. Lukei, and G. Fernndez. The visual object tracking vot2016 challenge results. In *Proceedings of the European Conference on Computer Vision Workshop*, pages 191–217. Springer International Publishing, 2016. 7, 8

[20] M. Kristan, J. Matas, A. Leonardis, and M. Felsberg. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*, pages 564–586, 2016. 1, 7, 8

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2

[22] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *Proceedings of the European Conference on Computer Vision Workshop*, pages 254–265, 2014. 1, 2

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[24] A. Lukei, T. Voj, L. ehovin, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[25] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, December 2015. 1, 2, 3, 5, 6, 7

[26] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5388–5396, 2015. 1, 2

[27] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 1, 2, 6

[28] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016. 2, 7

[29] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang. Object tracking via dual linear structured svm and explicit feature map. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4266–4274, 2016. 6, 7

[30] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015. 3

[31] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015. 3

[32] 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*, June 2016. 2, 3, 6, 7

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 91–99, 2015. 2

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 6

[35] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M. H. Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2, 6, 7

[36] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2016. 1, 3, 6, 7

[37] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 3, 6, 7

[38] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 689–692. ACM, 2015. 6

[39] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015. 6

[40] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017. 2, 3, 4

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

[42] Y. Wu, J. Lim, and M. H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(9):1834–1848, 2015. 1, 6, 8

[43] M. Zhang, J. Xing, J. Gao, and W. Hu. Robust visual tracking using joint scale-spatial correlation filters. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1468–1472. IEEE, 2015. 6

[44] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6, 7

[45] W. Zhang, P. Srinivasan, and J. Shi. Discriminative image warping with attribute flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2393–2400. IEEE, 2011. 3

[46] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3

[47] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[48] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang. Uct: Learning unified convolutional networks for real-time visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Oct 2017. 2, 5, 6, 7