Feature Integration with Adaptive Importance Maps for Visual Tracking

Aishi Li, Ming Yang*, Wanqi Yang
Nanjing Normal University
liamgsal@gmail.com, myang@njnu.edu.cn, yangwq@njnu.edu.cn

Abstract

Discriminative correlation filters have recently achieved excellent performance for visual object tracking. The key to success is to make full use of dense sampling and specific properties of circulant matrices in the Fourier domain. However, previous studies don't take into consideration the importance and complementary information of different features, simply concatenating them. This paper investigates an effective method of feature integration for correlation filters, which jointly learns filters, as well as importance maps in each frame. These importance maps borrow the advantages of different features, aiming to achieve complementary traits and improve robustness. Moreover, for each feature, an importance map is shared by its all channels to avoid overfitting. In addition, we introduce a regularization term for the importance maps and use the penalty factor to control the significance of features. Based on handcrafted and CNN features, we implement two trackers, which achieve a competitive performance compared with several state-of-the-art trackers.

1 Introduction

Visual tracking is one of the fundamental problems in computer vision and has numerous applications such as surveillance, robotic services and so on. Given the bounding box in the first frame, its task is to estimate the trajectory of a target in an image sequence. In the absence of prior information, it needs to handle various challenging problems, *i.e.,* deformations, occlusions, while maintaining tracking speed. Therefore, although significant progress has been made in recent years, visual tracking remains a challenging problem.

Here, we focus on the most general scenario of short-term, model-free tracking. Short-term means that tracker does not perform re-detection no matter the model drifts or not. Model-free is the main challenge where no prior information regarding the category and the appearance of a target but the bounding box in the first frame is available.

Most existing trackers focus on either the model for detection or feature representation. Discriminative correlation filters (DCF) for visual tracking [Bolme *et al.*, 2010] have been very popular due to robustness to the change in appearance and the superior computation. Dense sampling is the key to success of DCF, which is performed by circular shifts. However, these circular shifts are not equal to the actual translations. This results in boundary effects, which severely degrade the DCF's performance. There are two main solutions of boundary effects at present. One of them is to introduce a spatial regularization component (SRDCF) [Danelljan *et al.*, 2015b], which suffers from low frames per second. Another is to learn from the cropped patch [Kiani Galoogahi *et al.*, 2015; 2017], which can make the filter discriminative to the background.

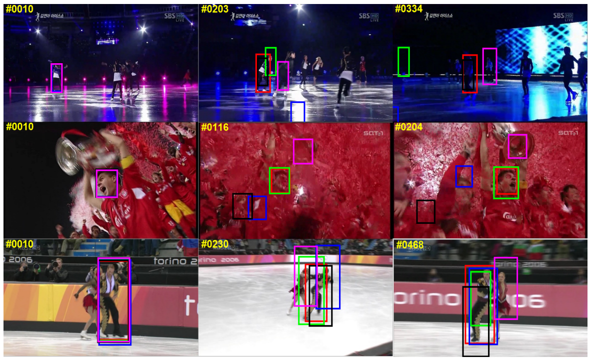

Figure 1: Qualitative results for our approach CFWFI (red), BACF (green), CSR-DCF (blue), ECO-HC (black), Staple (pink).

Feature representation is critical to the performance of visual tracking [Wang *et al.*, 2015]. DCF uses the grayscale intensity at the early stage because it cannot be applied to multi-channel. To improve the performance, the DCF formulation has been extended to multi-channel [Kiani Galoogahi *et al.*, 2013; Henriques *et al.*, 2015], which makes it possible to use more effective representation such as HOG, color features and deep features. At present, most of handcrafted features based trackers use color names and HOG as feature representation. For the better accuracy, DCF begins to use hierarchical convolutional features, sacrificing the tracking speed.

However, most trackers don't take into consideration importance and complementary information of features. Specifically, several recent trackers [Wang *et al.*, 2017; Lukezic *et al.*, 2017] are based on HOG features and color features, and simply concatenate them. As observed, HOG features focus on gradients of the target, which can describe the target's shape and are robust to illumination, while color features can extract features from homogeneous regions and show robustness to deformation. Therefore, these trackers don't exploit the advantages of every feature, just viewing all features as a multichannel feature.

*Corresponding author.

In order to address this limitation, we propose feature integration with adaptive importance maps for DCF (CFWFI). The method is based on BACF [Kiani Galoogahi *et al.*, 2017] which learns from background to deal with boundary effects. The main contributions of our work are as follows:

- In every frame, we jointly learn adaptive filters and importance maps which are used to achieve complementary traits. All channels of the same feature share an importance map, instead of learning an importance map for every channel, which maybe leads to overfitting.
- The adaptive importances of features are controlled by the penalty factors of importance maps. Specifically, compared with the grayscale intensity, HOG has a lower penalty factor for the importance map.
- We evaluate our method on OTB13 and OTB15 [Wu *et al.*, 2013; 2015] datasets. The results demonstrate that our method achieves competitive accuracy compared to the state-of-the-art DCF (Figure 1). Moreover, the handcrafted feature based tracker maintains a speed of 22 frames per second on an i5 CPU.

2 Related Work

2.1 The Recent Improvements of DCF

Due to the impressive speed of MOOSE [Bolme *et al.*, 2010], DCF has attracted wide attention. To improve the D-CF's performance, researchers take into consideration multichannel features [Kiani Galoogahi *et al.*, 2013; Henriques *et al.*, 2015], adaptive scale [Danelljan *et al.*, 2014a; Li and Zhu, 2014], nonlinear kernel [Henriques *et al.*, 2015; Tang and Feng, 2015], boundary effects [Danelljan *et al.*, 2015b; Kiani Galoogahi *et al.*, 2015; 2017], ensemble methods [Zhang *et al.*, 2017a; Zhang and Suganthan, 2017] respectively.

Inherent boundary effects of DCF severely degrade the tracking performance. To address this problem, SRDCF [Danelljan *et al.*, 2015b] introduces a spatial regularization depending on the spatial location to penalize the filter, which suffers from the slow speed. Context-aware CF [Mueller *et al.*, 2017], in addition to using the predicted outcomes as the training samples, imposes explicitly that filter's responses to the surrounding background equal zero. CFLB [Kiani Galoogahi *et al.*, 2013] crops the shifted patch to learn from the background. Afterwards, Hamed *et al.* [2017] extend it to the multi-channel, maintaining the real-time tracking and excellent accuracy.

2.2 Feature Representation of DCF

As is shown in [Wang *et al.*, 2015], feature representation plays the most critical role in tracking. Since the DCF formulation is extended to the multichannel, researchers begin to adopt more powerful handcrafted features to improve the tracking performance. KCF [Henriques *et al.*, 2015] works on HOG descriptors and achieves more superior accuracy compared with CSK [Henriques *et al.*, 2012]. However, when extending the linear kernel to the Gaussian kernel, KCF solely achieves a slight improvement. And Multi-kernel CF [Tang and Feng, 2015] attempts to take advantage of the invariance-discriminative power spectrums of various features to further improve the performance.

Besides HOG, CNs [Danelljan *et al.*, 2014b] and DAT [Possegger *et al.*, 2015] make color features attract more attention due to the favorable robustness to appearance's variance. Since observing the phenomenon that models based on color features can cope well with variation in shape but suffer when illumination, Staple [Bertinetto *et al.*, 2016a] combines HOG based DCF with color histogram based regression in order to make full use of complementary traits. However, to maintain real-time speed, Staple solves two independent ridge regression problems.

Several recent research [Lukezic *et al.*, 2017; He *et al.*, 2017] pays attention to the spatial and channel reliability. CSR-DCF [Lukezic *et al.*, 2017] based on HOG and color names learns separately the spatial reliability map, filters and channel reliabilities. The spatial reliability map is estimated using the output of a graph labeling problem. Filters are computed under the assumption of independent feature channel. And channel reliabilities depend on the maximum responses of channels. CF with weighted convolution [He *et al.*, 2017] extracts features from different layers of the CNN in order to learn filters. Weights in convolution responses represent the significance of layers. And then, the weighted convolution responses from all filters are summed to produce the final confidence. As other deep features based trackers, it also has a slow speed.

3 DCF tracking

We first review the conventional DCF. It extracts features $x \in \mathbb{R}^{DMN}$ from the predicted location of the target in the current frame to learn a correlation filter $h \in \mathbb{R}^{DMN}$ used to locate the target in the next frame, where MN represents the size of vectorized features, D denotes the number of channels or dimensions. x and h are represented by concatenating their channels $x_l \in \mathbb{R}^{MN}$, where l is the lth channel, $x = [x_1^T, x_2^T, ..., x_D^T]^T$ and $h = [h_1^T, h_2^T, ..., h_D^T]^T$. The key to DCF's high performance is dense sampling where circular shifts approximate the actual translations in an image. The objective function is to minimize the sum of squared error between the actual output and the desired output, which is ridge regression with a closed form solution.

$$\min_h \frac{1}{2}||\sum_{l=1}^{D} h_l \star x_l - y||^2 + \frac{\lambda}{2}||h||^2 \qquad (1)$$

where $y \in \mathbb{R}^{MN}$ is the vectorized 2D Gaussian response centered on the target in training image. The symbol $\star$ denotes the circular correlation. Due to Parseval's theorem and convolution theorem, ridge regression can be optimized quickly in the frequency domain.

$$\min_h \frac{1}{2}||\sum_{l=1}^{D} \hat{h}_l^* \odot \hat{x}_l - \hat{y}||^2 + \frac{\lambda}{2}||h||^2 \qquad (2)$$

where $\hat{h}_l, \hat{x}_l$ and $\hat{y}$ represents discrete Fourier transforms (DFT), *i.e.,* $\hat{h}_l = Fh_l, \hat{x}_l = Fx_l$ and $\hat{y} = Fy$, F denotes a DFT matrix. $\hat{h}_l^*$ indicates complex conjugate. The symbol $\odot$ denotes element-wise multiplication. Equation (2) can be solved efficiently in the Fourier domain.

$$\hat{h}_l = \frac{\hat{y}^* \odot \hat{x}_l}{\sum_{l=1}^{D} \hat{x}_l^* \odot \hat{x}_l + \lambda} \qquad (3)$$

And detection is also computed in the frequency domain.

$$r = \mathcal{F}^{-1}(\sum_{l=1}^{D} \hat{h}_l^* \odot \hat{x}_l) \qquad (4)$$

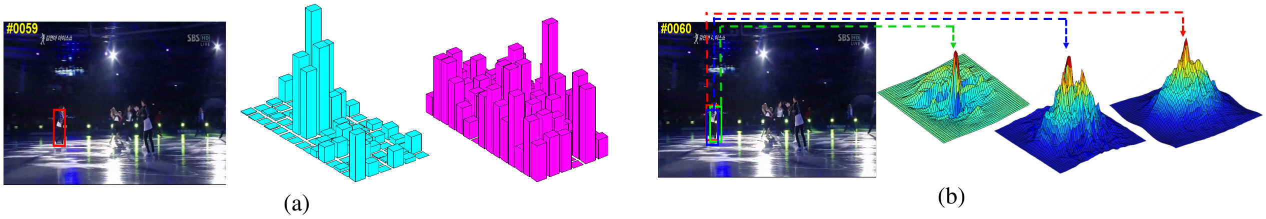

Figure 2: (a) Visualization of importance maps from the bounding box. The cyan map (for HOG) seems sparse and the magenta map (for the intensity) seems smooth, which indicates that HOG prefers to detect high-frequency features, while the intensity can extract features from homogeneous regions. (b) Illustration of responses from different features. The red is from HOG and the intensity, the green is from the intensity and the blue is from HOG. With the help of importance maps, our approach (red) borrows the advantages of HOG and intensity, provides a unimodal response and predicts a more accurate location.

where r denotes the response, $\mathcal{F}^{-1}$ represents the inverse of DFT.

4 Feature Integration with Adaptive Importance Maps

In this section, we first present the problem formulation of feature integration with adaptive importance maps. Then, we deduce an optimization algorithm with ADMM.

4.1 Problem Formulation

Feature representation is crucial to a tracker. Proper features can dramatically improve the tracking performance. Therefore, we use adaptive importance maps to improve the tracking performance. They take into consideration both the complementary traits and the importance of features. Specifically, $m \in \mathbb{R}^{MN \times K}$ denotes the importance maps and K indicates the number of features.

As observed, HOG features focus on gradients of the target, which can describe the target's shape and are robust to illumination, while color features can extract features from homogeneous regions and are robust to deformation. Also, the observation can be found in Figure 2(a) (the cyan map seems sparse and the magenta map seems smooth). In addition, in order to avoid overfitting, an importance map is shared by all channels of the same feature, instead every channel has its own importance map. Therefore, importance maps can achieve the complementary knowledge by laying stress on the focuses of every feature (see Figure 2(b)). However, for the sake of simplicity, importance maps are used for filters in our formulation.

Specifically, given a set of features $x \in \mathbb{R}^{DMN}$ extracted from an image, x euqals $[x^1; x^2; \ldots; x^K]$, $x^k \in \mathbb{R}^{D_k MN}$ represents the kth features. D_k denotes the dimension of the kth feature and D equals the sum of D_k. $x_{k,l} \in \mathbb{R}^{MN}$ denotes the lth channel of the kth feature and $h_{k,l} \in \mathbb{R}^{MN}$ represents the filter of $x_{k,l}$. To be simple, we convert the vector m_k to a diagonal matrix $diag(m_k) \in \mathbb{R}^{MN \times MN}$, where m_k is the importance map of the kth feature. We learn a filter $h \in \mathbb{R}^{DMN}$ used to locate the target in next frame by the following objective function:

$$\min_{h,m} \frac{1}{2}||y - \sum_{k=1}^{K}\sum_{l=1}^{D_k}(diag(m_k)h_{k,l}) \star x_{k,l}||^2 + \frac{\lambda}{2}||h||^2 \tag{5}$$

As we all know, since having the different discriminative ability, features are not equally important. Instead of weighted response sum [He *et al.*, 2017], we use the importance map to indicate the significance of a feature. The greater an importance map's magnitude is, the more important corresponding feature is. Hence, we introduce the L_2 regularization for importance maps, where $\alpha \in \mathbb{R}^K$ is the penalty coefficient of regularization. α controls the significance of every feature. Specifically speaking, when the value of α_k is small, the magnitude of m_k will be large which means the corresponding feature is more important. The complete formulation is as follows:

$$\begin{aligned}\min_{h,m} E(h,m) = \frac{1}{2}||y - \sum_{k=1}^{K}\sum_{l=1}^{D_k}(diag(m_k)h_{k,l}) \star x_{k,l}||^2 \\ + \frac{\lambda}{2}||h||^2 + \sum_{k=1}^{K}\frac{\alpha_k}{2}||m_k||^2 \\ s.t. \quad m \succeq 0\end{aligned} \tag{6}$$

where $m \succeq 0$ represents that every element in m is greater than or equal to zero.

4.2 Optimization Algorithm

To improve the computation efficiency, Equation (6) is transformed to the frequency domain according to convolution theorem and Parseval's theorem. Moreover, for the sake of simplicity, we introduce a variable $g \in \mathbb{R}^{DMN}$ to remove filters with important maps, where $g_{k,l} \in \mathbb{R}^{MN}$ represents the lth filter with the importance map in the kth feature. $x \in \mathbb{R}^{DMN}$ is transformed to $Z \in \mathbb{R}^{MN \times DMN}$ by diagonalization. And the primal objective function Equation (6) can be rewritten by vectorization as follow:

$$\begin{aligned}\min_{\hat{g},h,m} E^{'}(\hat{g},h,m) = \frac{1}{2}||\hat{y} - \hat{Z}\hat{g}||^2 + \frac{\lambda}{2}||h||^2 + \sum_{k=1}^{K}\frac{\alpha_k}{2}||m_k||^2 \\ s.t. \quad \hat{Z} = [diag(\hat{x}_{1,1}), ..., diag(\hat{x}_{1,D_1}), ..., diag(\hat{x}_{K,D_K})], \\ \hat{g}_{k,l} = Fdiag(m_k)h_{k,l}, \quad m_k \succeq 0\end{aligned} \tag{7}$$

Employing augmented Lagrangian, we get Equation (8) where $\xi_{k,l} \in \mathbb{R}^{MN}$ is the multiplier of $g_{k,l}$, $\xi \in \mathbb{R}^{DMN}$ denotes the Lagrange multiplier and μ denotes the penalty factor.

$$\begin{aligned}\min_{\hat{g},h,m} L(\hat{g},h,m) = \frac{1}{2}||\hat{y} - \hat{Z}\hat{g}||^2 + \frac{\lambda}{2}||h||^2 + \sum_{k=1}^{K}\frac{\alpha_k}{2}||m_k||^2 \\ + \sum_{k=1}^{K}\sum_{l=1}^{D_k}\hat{\xi}_{k,l}^T[\hat{g}_{k,l} - Fdiag(m_k)h_{k,l}] \\ + \frac{\mu}{2}\sum_{k=1}^{K}\sum_{l=1}^{D_k}||\hat{g}_{k,l} - Fdiag(m_k)h_{k,l}||^2\end{aligned} \tag{8}$$

Equation (8) can be solved with alternating direction method of multipliers (ADMM). Therefore, it can be divided into three subproblems. We first solve the subproblem $\hat{g}$, which is applied to detect the target in next frame.

Subproblem $\hat{g}^* = \arg\min_{\hat{g}} L(\hat{g}, h, m)$**:** Similar to ridge regression, this problem has a closed solution $\hat{g}^* = (\hat{Z}^T\hat{Z} + \mu I)^{-1}(\hat{Z}^T\hat{y} - \hat{\xi} + \mu\hat{h}^{'})$, where $\hat{h}^{'} \in \mathbb{R}^{DMN}$, $\hat{h}^{'}_{k,l} = Fdiag(m_k)h_{k,l}$ and I is the identity matrix. Observing that $\hat{x}$ is sparse banded [Kiani Galoogahi *et al.*, 2013], we can use Sherman-Morrison formula to speed up avoiding the inverse operation. Just as BACF's optimization [Kiani Galoogahi *et al.*, 2017],

$$\hat{g}^*(t) = \frac{1}{\mu}*[I - \frac{\hat{Z}(t)\hat{Z}(t)^T}{\mu + \hat{Z}(t)^T\hat{Z}(t)}]*[\hat{y}(t)\hat{Z}(t) - \hat{\xi}(t) + \mu\hat{h}^{'}(t)] \tag{9}$$

where $t = [1, 2, ..., MN]$ and t represents a position in the filters or features. And $\hat{g}(t) = conj([\hat{g}_1(t), ..., \hat{g}_D(t)]^T)$, $\hat{Z}(t) = [\hat{x}_1(t), ..., \hat{x}_D(t)]^T$, $\hat{\xi}(t) = [\hat{\xi}_1(t), ..., \hat{\xi}_D(t)]^T$, $\hat{h}^{'}(t) = [\hat{h}^{'}_1(t), ..., \hat{h}^{'}_D(t)]^T$. $conj$ represents conjuate.

Subproblem m**:** Observing that the objective function w.r.t. m is composed of K channels, $m_k \in \mathbb{R}^{MN}$ is independent to each other, which can be solved separately.

$$\min_m \sum_{k=1}^{K} \frac{\alpha_k}{2}||m_k||^2 + \sum_{k=1}^{K}\sum_{l=1}^{D_k} \xi_{k,l}^T[g_{k,l} - diag(m_k)h_{k,l}] + \frac{\mu}{2}\sum_{k=1}^{K}\sum_{l=1}^{D_k}||g_{k,l} - diag(m_k)h_{k,l}||^2 \tag{10}$$

$$m_k^* = \frac{\mu\sum_{l=1}^{D_k} h_{k,l} \odot g_{k,l} + \sum_{l=1}^{D_k} \xi_{k,l} \odot h_{k,l}}{\alpha + \mu\sum_{l=1}^{D_k} h_{k,l} \odot h_{k,l}} \tag{11}$$

If an element of m_k is less than zero, it will be set to zero.

Subproblem h**:** The objective function of h is a quadratic programming problem. And every element $h_{k,l}$ in h is independent, which can be solved easily.

$$\min_h \frac{\lambda}{2}||h||^2 + \sum_{k=1}^{K}\sum_{l=1}^{D_k} \xi_{k,l}^T[g_{k,l} - diag(m_k)h_{k,l}] + \frac{\mu}{2}\sum_{k=1}^{K}\sum_{l=1}^{D_k}||g_{k,l} - diag(m_k)h_{k,l}||^2 \tag{12}$$

$$h_{k,l}^* = \frac{m_k \odot \xi_{k,l} + \mu m_k \odot g_{k,l}}{\lambda + \mu m_k \odot m_k} \tag{13}$$

After estimating $\hat{h}^{'}_{k,l} = Fdiag(m_k)h_{k,l}$, the Lagrange multiplier is updated as

$$\hat{\xi}^t = \hat{\xi}^{t-1} + \mu(\hat{g}^t - \hat{h}^{'t}). \tag{14}$$

For updating the penalty factor μ, the standard scheme is applied as following:

$$\mu^t = \min\{\mu_{max}, \beta\mu^{t-1}\}. \tag{15}$$

Algorithm 1 CFWFI tracking algorithm

```
Require:
    Image $I_t$, location $p_{t-1}$, filter $g_{t-1}$, model $Z_{model}^{t-1}$
Ensure:
    Position $p_t$, updated model $Z_{model}^t$;
Extract features $Z^t$ centered on $p_{t-1}$;
Prediction: $p_t = \arg\max_p \mathcal{F}^{-1}(\hat{Z}^t\hat{g}^{t-1})$;
Extract features $Z^t$ centered on $p_t$;
Online update: $\hat{Z}_{model}^t = (1-\eta)\hat{Z}_{model}^{t-1} + \eta\hat{Z}^t$;
Initial: $h \Leftarrow$the cropped filter based on KCF, $m \Leftarrow 0$;
repeat
    Sequentially estimate $\hat{g}_t, m, h$ with Equations (9), (11), (13);
    Sequentially estimate $\hat{\xi}, \mu$ with Equations (14), (15);
until Maximum iteration;
return $\hat{g}_t, Z_{model}^t$.
```

It is noteworthy that initial values of h come from KCF's solutions. And Similar to BACF [Kiani Galoogahi *et al.*, 2017], we crop the center patch of h, other's values are assigned to zero. What's more, compared with the unitary DFT transform, the results of fft2 implemented by Matlab are $\sqrt{MN}$ times.

Instead of updating filters, we use linear interpolation $\hat{Z}_{model}^t = (1-\eta)\hat{Z}_{model}^{t-1} + \eta\hat{Z}^t$ to update target's model, where η is the online adaptation rate. A predicted location is the position with maximum value, $p_t = \arg\max_p \mathcal{F}^{-1}(\hat{Z}^t\hat{g}^{t-1})$. The whole algorithm is illustrated in Algorithm 1.

5 Experiments

To validate the effectiveness of proposed approach, we implement CFWFI based on handcrafted features and DeepCFWFI which is achieved by adding CNN features to CFWFI. Then we evaluate our trackers on the OTB13 and OTB15 [Wu *et al.*, 2013; 2015] benchmark datasets and compare them with some state-of-the-art methods.

5.1 Implementation Details

For real-time tracking, CFWFI's features are the grayscale intensity and HOG features with 31 channels. HOG features use 4*4 cell size to extract from an image patch. The area of image patch used to extract features is proportional to the area of the target bounding box. We set the region area to 5*5 times the target area and make it square. Samples extracted from the region are multiplied by a Hann window due to the DFT of samples. And this action ensures that tracker puts more emphasis near the center of the target. To learn and detect quickly, the maximum sample size is set to 50*50. Similar to SAMF's way [Li and Zhu, 2014] to deal with the scale variations, the number of scales is set to 5 with a scale-step of 1.01. The target output of correlation filter is a 2D Gaussian shaped response with the standard deviation of $\sqrt{wh}/16$, where w and h are the size of the target. The regularization factor λ of the filter is set to 0.01. In addition, the regularization factor α for importance maps is $[0.5, 0.01]$. The former is used for the grayscale intensity and the latter is used for HOG features. The learning rate η of updating the model is 0.013. Parameters of the alternating direction multiplier method are

	features	d_{ims}	OTB13	OTB15	FPS
CFWFI	HOG+gray	2	0.695	0.641	22
CFWFIcn	HOG+cn	2	0.671	0.624	21
CFWFIims	HOG+gray	32	0.641	0.618	22
CFWFInoims	HOG+gray	1	0.659	0.621	22
BACFgray	HOG+gray	0	0.657	0.622	23

Table 1: Characteristics of trackers and comparison of AUC scores. d_{ims} indicates the dimension of importance maps. CFWFInoims's importance maps are constants, which are composed of zero and one. The top 2 values are highlighted in red, green.

set to $\mu = 1$ and $\beta = 10$ where the former is the penalty factor and the latter is the update rate. We empirically find that ADMM with 2 iterations can achieve the solution close to the optimal, thus set the number of iteration to 2. Our Matlab R2014a implementation runs on an Intel i5 CPU PC with 4 GB memory.

Moreover, following DeepSRDCF [Danelljan *et al.*, 2015a], we add 96 channel features from the initial convolutional layer to CFWFI to implement DeepCFWFI. Learning rate η, regularization factor λ and α for CNN features are set to 0.009, 0.0001 and 0.05, respectively. Our Matlab R2015b implementation runs on an Intel i7 CPU PC and TITAN X.

5.2 Analyses of CFWFI

Here, we compare CFWFI with its variants, in order to demonstrate the effect of the proposed feature integration with adaptive importance map for visual tracking. We first investigate the consequence of removing adaptive importance maps (CFWFInoims), which is equivalent to the condition that the importance map is a constant matrix composed of zero and one. In addition, we compare the proposed CFWFI with BACFgray based on HOG and the grayscale intensity. What's more, to validate the previous suppose that overfitting occurs when all channels have their own importance map, we change CFWFI to CFWFIims. At the same time, we investigate the effect of integration of different features. We compare CFWFI with CFWFIcn. The former's features are composed of HOG and the grayscale intensity, the latter's features consist of HOG and color names. Table **??** shows characteristics of compared trackers and tracking performance of on OTB13 and OTB15. Figure 3 shows the success plots on OTB13 and OTB15, where numbers in legend represent the AUC.

As shown in Table **??**, frames per second of these handcrafted features based trackers are close to each other. Trackers with 20fps can be qualified for the real-time application. And, the success rate indicates that CFWFI achieves the best accuracy and robustness on OTB13 and OTB15, which benefits from the proposed feature integration with adaptive importance maps. CFWFInoims achieves a poorer performance than CFWFI, without adaptive importance maps. It validates the suppose that adaptive importance maps are helpful for feature integration to get complementary information. CFWFIims is the poorest tracker, because it learns an importance map for every channel, which leads to overfitting. Compared with CFWFIcn, CFWFI has a superior performance, which shows that feature integration with HOG and the grayscale intensity is more effective. The reason may be that color names have more channels and show the variance of an instance. Thus the importance map can not find the complementary traits effectively.

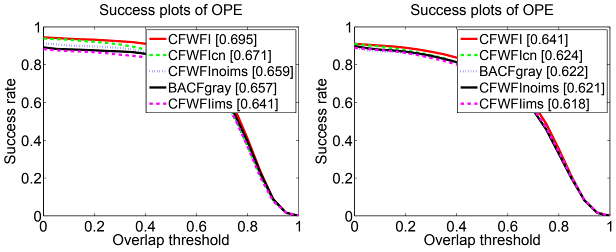

Figure 3: Success plots comparing CFWFI with variants on OTB13 (left) and OTB15 (right).

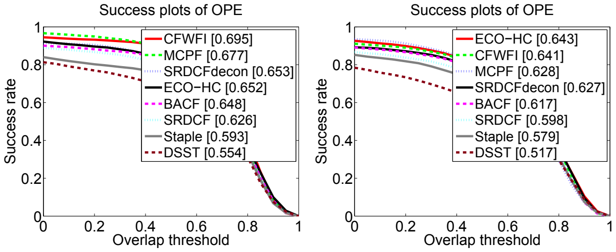

Figure 4: Success plots comparing CFWFI with handcrafted feature based trackers on OTB13 (left) and OTB15 (right).

5.3 Comparison with Handcrafted Feature based Trackers

Here, we compare our proposed CFWFI with handcrafted feature based trackers, including MCPF [Zhang *et al.*, 2017b], SRDCFdecon [Danelljan *et al.*, 2016a], ECO-HC [Danelljan *et al.*, 2017], BACF [Kiani Galoogahi *et al.*, 2017], SRDCF [Danelljan *et al.*, 2015b], Staple [Bertinetto *et al.*, 2016a], DSST [Danelljan *et al.*, 2014a] on the OTB13 and OTB15.

Figure 4 illustrates the success plots on OTB13 and OTB15, where CFWFI achieves the best AUC 0.695 on OTB13 and AUC 0.641 on OTB15, while maintaining real-time speed. Our method outperforms BACF's success rate of 0.648 on OTB13 and 0.617 on OTB15. These results demonstrate the effectiveness of feature integration with adaptive importance maps. In addition, Staple has poorer AUC scores compared with CFWFI, which learns filters from two independent ridge regressions in order to get complementary traits from HOG and color histogram. This indicates that our method is more effective to jointly learn adaptive importance maps and filters. MCPF provides AUC scores of 0.677 and 0.628, which gets a great performance at the cost of speed. ECO-HC obtains competitive AUC scores of 0.652 and 0.643. Different from SRDFC, our method learns from the background to address boundary effects, which has greater AUC scores and a faster speed than SRDCF and SRDCFdecon.

Attribute based Evaluation

We perform an attribute based evaluation of our approach on OTB13. Figure 5 shows success plots of six different attributes, including deformation, occlusion, illumination variation, background clutter, in-plane rotation, out-of-plane rotation. Our approach outperforms other trackers on these attributes. This indicates the effectiveness of feature integration with adaptive importance maps on different attributes and how adaptive importance maps improve the performance of a tracker.

Specifically, in the case of deformation, Staple achieves an AUC score of 0.607, which uses the color histogram to address deformation. Our approach achieves a best AUC score

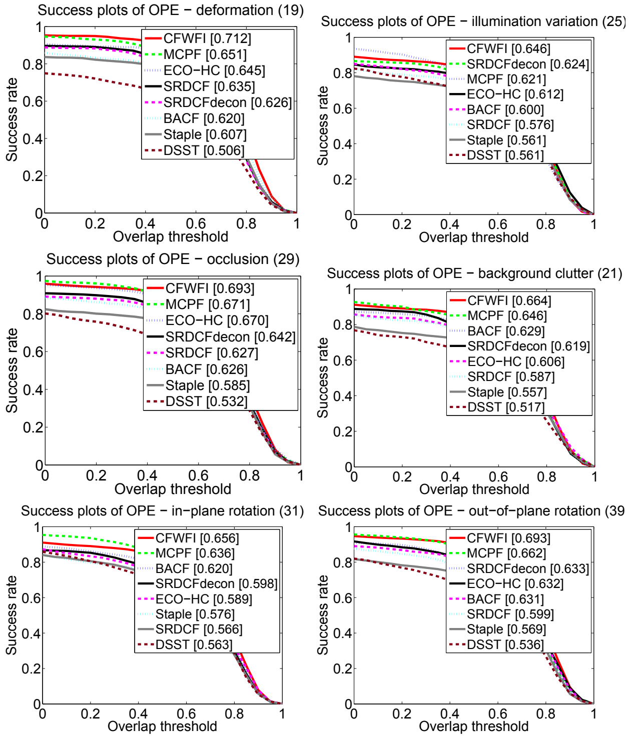

Figure 5: Success plots of attributes comparing CFWFI with state-of-the-art trackers on OTB13.

of 0.712, followed by MCPF with an AUC score of 0.651. In the case of illumination variant, CFWFI achieves the best AUC score of 0.646 followed by the AUC score of 0.624 over SRDCFdecon. These results demonstrate that importance maps can effectively address deformation and illumination (see Figure 6), because they can help tracker achieve complementary information to improve robustness. In addition, because HOG focuses on gradients, the setting that HOG is more important in CFWFI improves the robustness to illumination variant. MCPF and ECO-HC provide the great results in the case of occlusion. Our method achieves an improvement of 0.022 over MCPF. As for background clutter, CFWFI provides a gain of 0.018 compared with MPCF and 0.035 compared with BACF. This benefits from the cropping operator coming from BACF, which can learn from the background. Eventually, CFWFI also provides the greater AUC scores on in-plane rotation and out-of-plane rotation, which achieves a gain of 0.02 and 0.031 compared with the second tracker. In addition to the online update, the fact that CFWFI can learn from the background contributes to the results.

5.4 Comparison with Deep Feature based Trackers

To validate the effectiveness of the proposed method, we follow DeepSRDCF [Danelljan *et al.*, 2015a] and implement DeepCFWFI by adding CNN based features to CFWFI. As shown in Figure 7, compared with C-COT [Danelljan *et al.*, 2016b], CFNet [Valmadre *et al.*, 2017], SiamFC [Bertinetto *et al.*, 2016b] and DeepSRDCF [Danelljan *et al.*, 2015a], DeepCFWFI achieves a competitive performance at a speed of 9 fps on an i7 CPU and TITAN X.

5.5 Analyses of Regularization Parameter α

We select sequentially α for HOG, gray and CN and analyze the effect of regularization parameter based on handcrafted features. Combining HOG with color features, the proposed tracker is more robust when HOG's α is less than CN's or gray's. It validates our assumption that HOG is more important than CN or gray. Furthermore, compared with using HOG and gray as features, adding CN to features sacrifices the speed of tracker and cannot significantly improve performance. Because with the help of the importance maps, both gray and CN extract features from homogeneous regions to cooperate with high-frequency features extracted by HOG.

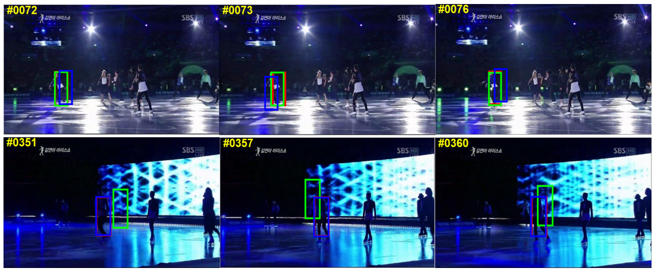

Figure 6: Illustration of predictions from different features. The red is from HOG and the instensity, the green is from the intensity and the blue is from the HOG. Our approach (red) provides an accurate prediction. (Top row) The intensity (green) shows robustness to deformation, while the HOG fails. (Bottom row) HOG (blue) shows robustness to illumination, while the intensity fails.

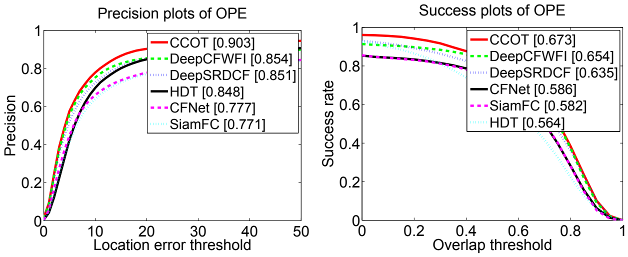

Figure 7: Precision plots and success plots comparing DeepCFWFI with deep feature based trackers on OTB15.

6 Conclusion

In this paper, we propose a novel feature integration with adaptive importance maps for visual tracking. To avoid overfitting, every feature has its own importance map which is shared by all channels of the same feature. Since importance maps can put emphasis on the focuses of every feature, tracker achieves the complementary traits. In addition, avoiding weighted response sum, our approach controls the significance of feature by the regularization factor of its importance map. We jointly learn importance maps and filters using ADMM and perform experiments on OTB13 and OTB15. Comparison with state-of-the-art trackers demonstrates the effectiveness of importance maps. Especially when deformation or illumination variant occurs, our method can use complementary information and address them effectively. And our approach achieves the competitive performance while maintaining the tracking speed.

Acknowledgments

This work is supported from the National Natural Science Foundation of China (Nos. 61603193, 61432008, 61272222), the Natural Science Foundation of Jiangsu Province (Nos. BK20171479, BK20161020, BK20161560) and Jiangsu Postdoctoral Science Foundation (No. 1701157B).

References

[Bertinetto *et al.*, 2016a] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, pages 1401–1409, 2016.

[Bertinetto *et al.*, 2016b] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865. Springer, 2016.

[Bolme *et al.*, 2010] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550. IEEE, 2010.

[Danelljan *et al.*, 2014a] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*. BMVA Press, 2014.

[Danelljan *et al.*, 2014b] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, pages 1090–1097, 2014.

[Danelljan *et al.*, 2015a] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCVW*, pages 58–66, 2015.

[Danelljan *et al.*, 2015b] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318, 2015.

[Danelljan *et al.*, 2016a] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, pages 1430–1438, 2016.

[Danelljan *et al.*, 2016b] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488. Springer, 2016.

[Danelljan *et al.*, 2017] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, July 2017.

[He *et al.*, 2017] Zhiqun He, Yingruo Fan, Junfei Zhuang, Yuan Dong, and HongLiang Bai. Correlation filters with weighted convolution responses. In *CVPR*, pages 1992–2000, 2017.

[Henriques *et al.*, 2012] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715. Springer, 2012.

[Henriques *et al.*, 2015] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[Kiani Galoogahi *et al.*, 2013] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Multi-channel correlation filters. In *ICCV*, pages 3072–3079, 2013.

[Kiani Galoogahi *et al.*, 2015] Hamed Kiani Galoogahi, Terence Sim, and Simon Lucey. Correlation filters with limited boundaries. In *CVPR*, pages 4630–4638, 2015.

[Kiani Galoogahi *et al.*, 2017] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, Oct 2017.

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

[Lukezic *et al.*, 2017] Alan Lukezic, Tomas Vojir, Luka Cehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *CVPR*, July 2017.

[Mueller *et al.*, 2017] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *CVPR*, pages 1387–1395, 2017.

[Possegger *et al.*, 2015] Horst Possegger, Thomas Mauthner, and Horst Bischof. In defense of color-based model-free tracking. In *CVPR*, pages 2113–2120, 2015.

[Tang and Feng, 2015] Ming Tang and Jiayi Feng. Multi-kernel correlation filter for visual tracking. In *ICCV*, pages 3038–3046, 2015.

[Valmadre *et al.*, 2017] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 5000–5008. IEEE, 2017.

[Wang *et al.*, 2015] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. Understanding and diagnosing visual tracking systems. In *ICCV*, pages 3101–3109, 2015.

[Wang *et al.*, 2017] Mengmeng Wang, Yong Liu, and Zeyi Huang. Large margin object tracking with circulant feature maps. In *CVPR*, July 2017.

[Wu *et al.*, 2013] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418, 2013.

[Wu *et al.*, 2015] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015.

[Zhang and Suganthan, 2017] Le Zhang and Ponnuthurai Nagaratnam Suganthan. Robust visual tracking via co-trained kernelized correlation filters. *Pattern Recognition*, 69:82–93, 2017.

[Zhang *et al.*, 2017a] Le Zhang, Jagannadan Varadarajan, Ponnuthurai Nagaratnam Suganthan, Narendra Ahuja, and Pierre Moulin. Robust visual tracking using oblique random forests. In *CVPR*, 2017.

[Zhang *et al.*, 2017b] Tianzhu Zhang, Changsheng Xu, and Ming-Hsuan Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, July 2017.