FLIPFLOP CORRELATION TRACKING WITH CONVOLUTION KERNELS NETWORKS

Hui He[1], *Bo Ma*[*1] *and Luoyu Qin*[2]

[1]Beijing Laboratory of Intelligent Information Technology, Beijing Institute of Technology, China
[2]China Academy of Space Technology, China

ABSTRACT

Correlation filter-based tracking methods have accomplished competitive performance on accuracy and robustness, but there is still a huge potential in choosing suitable features. Recently, Convolutional Kernel Networks (CKN), which provide a fast and simple procedure to approximate kernel descriptors, have been proposed and achieved state-of-the-art performance in many vision tasks. In this paper, we present an adaptive tracker which integrates the kernel correlation filters with multiple effective CKN descriptors. By adopting a FlipFlop scheme, the weights of different features can be adjusted in the process of tracking to get better performance. Extensive experimental results on the OTB-2013 tracking benchmark show that our approach performs favorably against some representative state-of-the-art tracking algorithms.

Index Terms— correlation tracking, convolutional kernel networks, adaptive multiple features

1. INTRODUCTION

Visual tracking, whose goal is to estimate the states of the target in the subsequent frames[1], plays a critical role in numerous computer vision applications such as surveillance, robotics and behavior analysis. Although decades of research have been studied in this field, it is still a challenging and interesting task due to several complication factors, such as background clutter, illumination variation, partial occlusions and deformation.

From the perspective of the foreground and background information usage, the mainstream tracking methods can be categorized into generative ones and discriminative ones. Generative trackers focus on establishing robust appearance models of the target by using templates or subspaces and performing tracking by searching the best-matching windows [2, 3]. While discriminative trackers often construct online classifiers which aim to distinguish the target from its backgrounds [4, 5, 6, 7]. It has been proved that background information involved in discriminative methods is advantageous in effective tracking [8].

In particular, the correlation filter-based discriminative trackers have made significant achievements recently and attracted much attention [9, 10, 11, 12]. As proposed in [13], by expanding single-channel filter to multi-channels and replacing original pixel values with Histogram of Oriented Gradients (HOG), Kernelized Correlation Filter (KCF) tracker is more competitive in performance than state-of-the-art trackers with high speed running at hundreds of frames-per-second. Considering that the HOG is a hand-crafted feature, it is necessary to extract more effective features for better performance.

Recently, Convolutional Kernel Networks (CKN) [14], as a simple convolutional neural network to approximate patch-based kernel descriptors [15], have been developed to provide a end-to-end image representation and demonstrated state-of-the-art performance in many vision tasks, such as classification [14] and image retrieval [16]. Although these semantic representations are shown to be very effective in categorizing and capturing original spatial details of objects [17], they are not the optimal representation for visual tracking. To achieve better performance, it is imperative to combine multiple features for best representation and separate foreground targets from the background clutters.

Decision-theoretic online learning (DTOL) [18] is a framework to dynamically allocate resources among some experts and capture learning problems proceeding in rounds, which is suitable for combining multiple responses to final decision in visual tracking. Hedge algorithm, which uses a set of experts to explain the observations regardless of how the observations are generated, is first proposed to solve the DTOL problem. The resource assignment of each expert depends on the cumulative loss of this expert and a learning rate parameter. However, Hedge algorithm cannot ensure the best prediction in various applications as the best learning rate cannot be obtained at all times [19]. AdaHedge [20] and FlipFlop [21] are both proposed by Tim et al. to overcome the drawback of the original Hedge algorithm. By dividing original learning problem to sub-problem, the learning rate parameter can be directly obtained by a part of the loss, which makes the decision more close to the optimal result. Considering the superiority of FlipFlop, in this paper, we use this

*Corresponding author: bma000@bit.edu.cn (Bo Ma). This work was supported in part by the National Natural Science Foundation of China (No. 61472036).

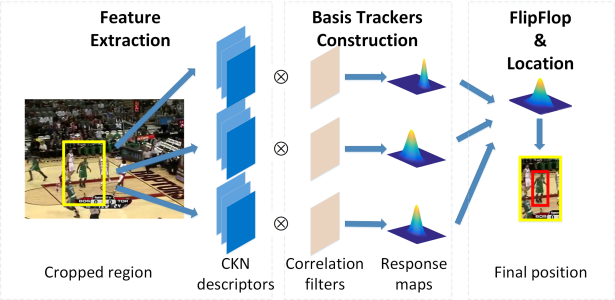

Fig. 1. The process of proposed tracking method.

method to combine all basis trackers for better results.

Contributions: The contributions of this work are three-fold.

(1) To the best of our knowledge, we are the first to introduce CKN into visual tracking, which adopts CKN descriptors with KCF framework as basis trackers.

(2) We propose an adaptive FlipFlop algorithm based on DTOL framework for visual tracking by considering the cumulative loss of basis trackers.

(3) We carry out extensive experiments on a large-scale benchmark dataset [1] with 51 challenging sequences to demonstrate the effectiveness of the proposed algorithm with comparisons to state-of-the-art trackers.

2. PROPOSED TRACKING ALGORITHM

2.1. Pipeline

As shown in Fig. 1, correlation filters integrate with CKN descriptors are considered as basis trackers, then adaptive FlipFlop algorithm which combines all basis trackers is used to predict the final position.

The first step of our method is extracting image features. Three pre-trained CKNs proposed in [16] are used to extract features from the cropped image region, which represent the target with different semantic. CKN-raw directly feeds the raw RGB patch to the network which captures the hue information but sensitive to environment illumination. PCA-whitening pre-processing consists in each sub-patch of CKN-white, which makes it more invariant to color. Taking the gradient along each spatial dimension with 1×1 sub-patches as the input for the network, CKN-grad is fully invariant to color.

Then, each feature map is used to construct basis trackers. As a generalization of HOG, CKN descriptors can be approximately considered as block-cyclic with unfixed block size. So we use an adaptive cell size in this paper to agree with the formation of KCF tracker. Similar to KCF [13], basis tracker model consists of the learned target appearance $\mathbf{x}$ and the transformed classifier coefficients

$$\alpha = \mathcal{F}^{-1}(\frac{\mathcal{Y}}{\kappa(\mathcal{X},\mathcal{X})+\lambda}), \tag{1}$$

where $\kappa(,)$ is kernel function, $\mathcal{X} = \mathcal{F}(\mathbf{x}), \mathcal{Y} = \mathcal{F}(\mathbf{y})$, $\mathcal{F}(.)$ denotes discrete Fourier transformation (DFT), $\mathcal{F}^{-1}$ denotes the inverse of DFT and $\mathbf{y}$ is a Gaussian shape label matrix. And the response map of each basis tracker is obtained by

$$C = \mathcal{F}^{-1}(\kappa(\mathcal{Z},\mathcal{X}) \odot \mathcal{F}(\alpha)), \tag{2}$$

where $\mathcal{Z} = \mathcal{F}(\mathbf{z})$ and patch $\mathbf{z}$ with the same size of $\mathbf{x}$ is cropped from the new frame, the symbol $\odot$ denotes element-wise product.

All response maps are finally combined to form a stronger tracker, which exploits the strength of different CKN descriptors for robust performance. In the t-th frame, the final target position is predicted by weighted decisions of all trackers

$$\begin{aligned} C_t^* &= \sum_{k=1}^{K} \omega_t^k C_t^k, \\ (x_t^*, y_t^*) &= \arg\max_{x,y} C_t^*(x,y), \end{aligned} \tag{3}$$

where ω_t^k is the weight of basis tracker k and $\sum_{k=1}^{K} \omega_t^k = 1$, C_t^k is the confidence map and $C_t^*(x,y)$ denotes the element at position (x,y) of response matrix C_t^*.

2.2. Adaptive FlipFlop

Following the DTOL framework, each expert's weight needs to be updated after the prediction of the ultimate target position, which depends on the loss that expert incur. In this work, the loss suffered by expert k at frame t is defined as:

$$l_t^k = C_t^k(x_t^*, y_t^*) - \max(C_t^k), \tag{4}$$

where $\max(C_t^k)$ returns the largest element of the matrix C_t^k.

Let $L_t^k = l_1^k + ... + l_t^k$ denotes the cumulative loss of expert k after t frames. The loss incurred by Hedge in frame t is $h_t = \boldsymbol{\omega}_t^\top \mathbf{l}_t$ ($\boldsymbol{\omega}_t = [\omega_t^1, ..., \omega_t^K]^\top, \mathbf{l}_t = [l_t^1, ..., l_t^K]^\top$) and the cumulative Hedge loss is $H_t = h_1 + ... + h_t$.

Learners performance is evaluated in terms of its regret, which is the difference between the cumulative Hedge loss and the cumulative loss of the best expert: $R_t = H_t - L_t^*$, where $L_t^* = \min_k L_t^k$.

Our goal is to minimize the regret after T frames, which crucially depends on learning rate η [20]. To this end, it turns out to be technically convenient to approximate h_t by the mix loss $m_t = -\frac{1}{\eta}\ln(\boldsymbol{\omega}_t \cdot e^{-\eta \mathbf{l}_t})$, which accumulates to $M_t = m_1 + ... + m_t$. Let $\delta_t = h_t - m_t$ denotes the mixability gap and $\Delta_t = \delta_1 + ... + \delta_t$ denotes the cumulation, so that the regret for Hedge may be decomposed as

$$R_t = H_t - L_t^* = M_t - L_t^* + \Delta_t. \tag{5}$$

As the cumulative mixability gap Δ_t is nondecreasing in t and can be observed online, it is possible to adapt the learning rate directly based on Δ_t [20]. Following the FlipFlop

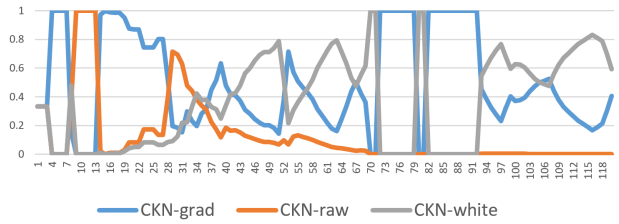

Fig. 2. The weights of CKN descriptors in sequence *Crossing*.

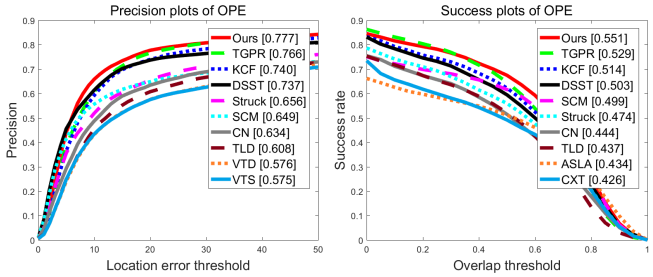

Fig. 3. Evaluation results on OTB-2013 database.

algorithm, the learning rate η_t now alternates between infinity, such that the algorithm behaves like Follow-the-Leader (FTL), and the AdaHedge value [20], which decreases as a function of the mixability gap accumulated over the rounds

$$\eta_t = \begin{cases} \eta_t^{\text{flip}} = \infty, & \text{if } t \in \overline{R}_t \\ \eta_t^{\text{flop}} = \frac{\ln K}{\underline{\Delta}_{t-1}}, & \text{if } t \in \underline{R}_t \end{cases}, \tag{6}$$

where $\overline{R}_t$ is the flip regime, which is the subset of times $\{1, ..., t\}$ and $\underline{R}_t$ is the flop regime, $\overline{R}_t \cup \underline{R}_t = \{1, ..., t\}$.

Then the weight and mix loss can be updated using this new learning rate η_t, for flip regime:

$$\begin{aligned} w_t^k &= \begin{cases} 1, & \text{if } L_t^k = L_t^* \\ 0, & \text{otherwise} \end{cases}, \\ m_t &= L_t^* - L_{t-1}^*, \end{aligned} \tag{7}$$

and for flop regime:

$$\begin{aligned} w_t^k &= \frac{w_1^k e^{-\eta_t L_{t-1}^k}}{\boldsymbol{\omega}_1^\top e^{-\eta_t \mathbf{L}_{t-1}}}, \\ m_t &= -\frac{1}{\eta_t} \ln(\boldsymbol{\omega}_t^\top \cdot e^{-\eta_t \mathbf{l}_t}). \end{aligned} \tag{8}$$

where $\boldsymbol{\omega}_1 = [1/K, .., 1/K]^\top$ and $\mathbf{L}_{t-1} = [L_{t-1}^1, ..., L_{t-1}^K]^\top$.

After that, new cumulative mixability gaps $\overline{\Delta}_t$ and $\underline{\Delta}_t$ can be obtained respectively by

$$\begin{aligned} \overline{\Delta}_t &= \begin{cases} \overline{\Delta}_{t-1} + \delta_t, & \text{if } t \in \overline{R}_t \\ \overline{\Delta}_{t-1}, & \text{if } t \in \underline{R}_t \end{cases}, \\ \underline{\Delta}_t &= \begin{cases} \underline{\Delta}_{t-1}, & \text{if } t \in \overline{R}_t \\ \underline{\Delta}_{t-1} + \delta_t, & \text{if } t \in \underline{R}_t \end{cases}, \end{aligned} \tag{9}$$

where $\delta_t = h_t - m_t$.

FlipFlop starts with an epoch of the flip regime until $\overline{\Delta}_t > (\varepsilon/\tau)\underline{\Delta}_t$ where ε and τ are scale parameters. At that point it switches to an epoch of the flop regime, and keeps using η_t^{flop} until $\underline{\Delta}_t > \tau\overline{\Delta}_t$. Then the process repeats with the next epochs of the flip and flop regimes [21].

Fig.2 illustrates the process of FlipFlop in sequence *Crossing*. It can be observed that the weight of CKN-raw works only for the first half of the sequence, and the weight of CKN-grad and CKN-white compete with each other in the whole video.

2.3. Update Scheme

During tracking, the target object may move through different lighting conditions, become occluded by other objects, or its pose or appearance may undergo significant changes. Therefore, filters need to adapt for robustness quickly.

For each basis tracker, we update its corresponding filter $(\alpha_t^k, \mathbf{x}_t^k)$ over time with the same learning rate τ by

$$\begin{aligned} \alpha_t^k &= \begin{cases} (1-\tau)\alpha_{t-1}^k + \tau\hat{\alpha}^k, & \text{if } r_k > \text{threshold} \\ \alpha_{t-1}^k, \text{otherwise} \end{cases}, \\ \mathbf{x}_t^k &= \begin{cases} (1-\tau)\mathbf{x}_{t-1}^k + \tau\hat{\mathbf{x}}^k, & \text{if } r_k > \text{threshold} \\ \mathbf{x}_{t-1}^k, \text{otherwise} \end{cases}, \end{aligned} \tag{10}$$

where $\hat{\alpha}^k, \hat{\mathbf{x}}^k$ are learned from the ultimate target position, r_k is the reliable scores of basis tracker k defined as $r_k = \frac{1}{1+\exp(-\max(C_t^k))}$.

Obviously, a small reliable score indicates the heavy occlusion, abrupt motion, scaling or sudden pose change of target in the current frame. Thus, for a score in the current frame smaller than a predefined threshold (0.55 in this work), we keep previous filters.

3. EXPERIMENTAL RESULTS

3.1. Experiment Setup

To evaluate the performance of our tracker, we conduct experiments on the benchmark dataset OTB-2013 proposed in [1], which includes 51 challenging image sequences. All sequences are categorized into 11 attributes based on different challenging factors, including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-view, background clutters, and low resolution.

To carry out comprehensive and fair comparisons, we compare our approach with all tracking algorithms reported in [1] (e.g., Struck [22], TLD [23], ASLA [24] and SCM [25]) and some state-of-the-art tracking methods including CN [12], KCF [13], TGPR [26] and DSST [27]. These trackers are evaluated using the source codes from the original

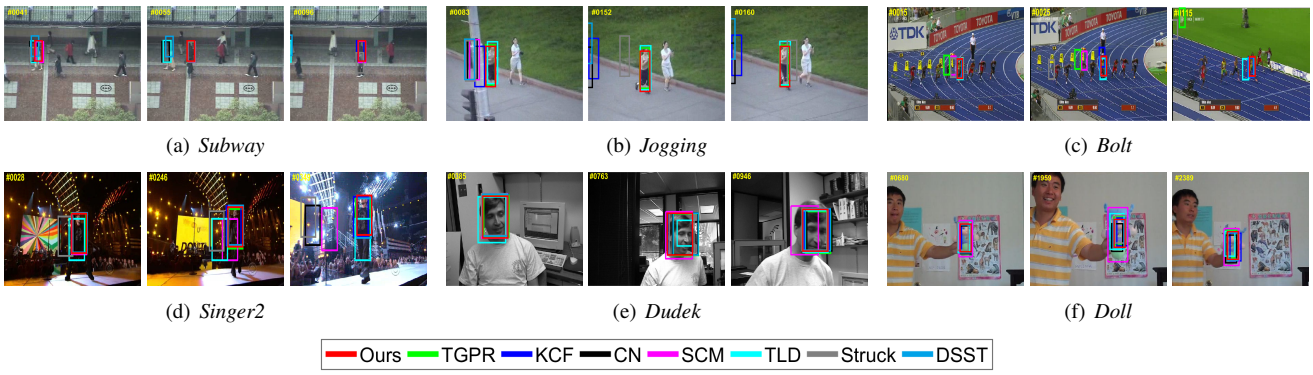

Fig. 4. Qualitative results of 8 trackers over sequences *Subway*, *Jogging*, *Bolt*, *Singer2*, *Dudek* and *Doll*.

authors and each is run with default parameters. Our method is implemented in MATLAB and all experiments are carried out on an Intel Core i5 processor with 3.1 GHz frequency and 6 GB RAM.

3.2. Quantitative and Qualitative Evaluation

Quantitative evaluation: Fig. 3 shows the precision plots and success plots of the trackers on 51 videos. Experimental results are reported using overlap success (OP) plots and center location error (CLE) plots. For precision plots, we rank the trackers according to the results at the error threshold of 20 pixels. For success plots, the trackers are ranked according to the AUC scores. The precision scores and AUC scores for each tracker are shown in the legend. Only top 10 of the trackers are displayed for clarity.

It can be observed that our method ranks 1st on precision rate and success rate, whose precision score is 0.777 and overlap rate is 0.551. In the precision plots, our algorithm performs 5% better than KCF and 5.4% better than DSST. We can also observe from the precision plots that our method performs slightly better than others when the location error threshold is smaller than 10 pixels. This is possibly because that the CKN descriptors can achieve a pixel-level similarity measure. In the success plots, our tracker outperforms TGPR by 5.8% and KCF by 7.2%. When given a specific overlap threshold (e.g., 0.5), our method still achieves the best performance.

Qualitative evaluation: As shown in Fig. 4(a)(b) , the targets in sequences *Subway* and *Jogging* are undergo heavy occlusion. In sequence *Subway*, a person is occluded by other people (e.g.,#41, #96). Only our method, TGPR, SCM and KCF can track the target stably. In the *Jogging* sequence, the left girl is occluded fully by the telegraph pole (e.g.,#73). Only our method and TLD can track the target successfully (e.g., #83, #92). Our method performs favorably because it employs an update scheme based on reliable scores.

Fig. 4(c)(d) illustrate some screenshots of tracking results in three challenging sequences where the target appearances undergo severe deformation. In the *Bolt* sequence, several objects appear on the screen with rapid appearance changes due to shape deformation and fast motion. Our method, CN and DSST algorithms can track the target stably. The TGPR, SCM, Struck and KCF methods suffer from severe drift at the beginning of the sequence (e.g.,#15, #25). The TLD algorithm drifts to the background at frame #115. The target in the *Singer2* sequence undergoes both deformation and illumination variations. Our method performs well in the whole sequence as adaptive FlipFlop scheme exploits the strength of different CKN descriptors, which is robustness to appearance variations.

To evaluate our method in more general cases, we select some sequences with in-plane or/and out-of-plane rotations. It can be observed from Fig. 4(e)(f) that rotation of the target makes it much more indistinguishable in a new frame and casts a more difficult problem in tracking. In sequence *Dudek*, the person rotates his head for about 360 degrees. TGPR, TLD, SCM and KCF sometimes drift away with occlusion or illumination variation (e.g., #385, #763, #946). Our method locks the target till the end with the correct scale. In the *Doll* sequence, SCM and TLD drift several pixels away from the target due to scale variation (e.g., #1959). Struck, KCF, CN and DSST drift away with fast motion blur and background change (e.g., #680, #2389). Only TGPR and our tracker successfully track the doll in the whole sequence.

4. CONCLUSION

To improve the performance of correlation filter-based tracking methods, we introduce a model which integrates KCF with multiple effective CKN descriptors under DTOL framework. With the loss of each basis tracker is considered and FlipFlop algorithm is adopted, the weights of different features can be adjusted in the process of tracking. Comprehensive experimental comparisons with the state-of-the-art algorithms on 51 challenging sequences demonstrate the effectiveness of the proposed tracking method.

5. REFERENCES

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[2] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[3] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR*. IEEE, 2012, pp. 1830–1837.

[4] Shai Avidan, "Support vector tracking," *TPAMI*, vol. 26, no. 8, pp. 1064–1072, 2004.

[5] Helmut Grabner, Michael Grabner, and Horst Bischof, "Real-time tracking via on-line boosting.," in *BMVC*, 2006, vol. 1, p. 6.

[6] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Visual tracking with online multiple instance learning," in *CVPR*. IEEE, 2009, pp. 983–990.

[7] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *T-CSVT*, vol. 24, no. 2, pp. 242–254, 2014.

[8] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.

[9] Zhe Chen, Zhibin Hong, and Dacheng Tao, "An experimental survey on correlation filter-based tracking," *arXiv preprint arXiv:1509.05520*, 2015.

[10] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*. IEEE, 2010, pp. 2544–2550.

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

[12] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *CVPR*, 2014, pp. 1090–1097.

[13] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[14] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid, "Convolutional kernel networks," in *NIPS*, 2014, pp. 2627–2635.

[15] Liefeng Bo, Xiaofeng Ren, and Dieter Fox, "Kernel descriptors for visual recognition," in *NIPS*, 2010, pp. 244–252.

[16] Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid, "Local convolutional features with unsupervised training for image retrieval," in *ICCV*, 2015, pp. 91–99.

[17] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox, "Object recognition with hierarchical kernel descriptors," in *CVPR*. IEEE, 2011, pp. 1729–1736.

[18] Yoav Freund and Robert E Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *EuroCOLT*. Springer, 1995, pp. 23–37.

[19] Shengping Zhang, Huiyu Zhou, Hongxun Yao, Yanhao Zhang, Kuanquan Wang, and Jun Zhang, "Adaptive normalhedge for robust visual tracking," *Signal Processing*, vol. 110, pp. 132–142, 2015.

[20] Tim V Erven, Wouter M Koolen, Steven D Rooij, and Peter Grünwald, "Adaptive hedge," in *NIPS*, 2011, pp. 1656–1664.

[21] Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen, "Follow the leader if you can, hedge if you must.," *JMLR*, vol. 15, no. 1, pp. 1281–1316, 2014.

[22] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *ICCV*. IEEE, 2011, pp. 263–270.

[23] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[24] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*. IEEE, 2012, pp. 1822–1829.

[25] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *CVPR*. IEEE, 2012, pp. 1838–1845.

[26] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing, "Transfer learning based visual tracking with gaussian processes regression," in *ECCV*. Springer, 2014, pp. 188–203.

[27] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*. BMVA Press, 2014.