Context Adaptive Visual Tracker in Surveillance Networks

Wei Feng[1], Minye Li[2], Yuan Zhou[3(✉)], Zizi Li[3], and Chenghao Li[3]

[1] Systems Engineering Research Institute of China State Shipbuilding Corporation, Beijing 100036, China
[2] Unit 61660 of PLA, Beijing 100089, China
[3] Tianjin University, Tianjin 300072, China
zhouyuan@tju.edu.cn

Abstract. CNN-based visual trackers has been successfully applied to surveillance networks. Some trackers apply sliding-window method to generate candidate samples which is the input of network. However, some candidate samples containing too much background regions are mistakenly used for target tracking, which leads to a drift problem. To mitigate this problem, we propose a novel Context Adaptive Visual tracker (CAVT), which discards the patches containing too much background regions and constructs a robust appearance model of tracking targets. The proposed method first formulates a weighted similarity function to construct a pure target region. The pure target region and the surrounding area of the bounding box are used as a target prior and a background prior, respectively. Then the method exploits both the target prior and background prior to distinguish target and background regions from the bounding box. Experiments on a challenging benchmark OTB demonstrate that the proposed CAVT algorithm performs favorably compared to several state-of-the-art methods.

Keywords: Visual tracking · Surveillance network

1 Introduction

Visual tracking is one of the most fundamental problems in computer vision with various applications such as surveillance and vehicle navigation. Despite great progress has been made over the past decade, it remains a challenging problem to design a robust tracker due to factors such as occlusion, illumination changes and geometric deformations, etc.

Recently, CNN-based discriminative tracking methods have been proven to be capable to achieve favorable tracking performance. Particularly, some trackers use sliding-window method to generate candidate samples which is the input of network. Although achieved the encouraging results both in accuracy and robustness, some candidate samples containing too much background regions

S. Han et al. (Eds.): AICON 2019, LNICST 286, pp. 374–382, 2019.
https://doi.org/10.1007/978-3-030-22968-9_33

are mistakenly used for target tracking, which leads to a drift problem. The reason is as follows.

At the initial frame of a video sequence, the patches of a tracking target are sampled from the region which is manually labeled as a target bounding box (a rectangular box). The patches of the tracking target are input to the network to extract the feature of the target, and then the classifier distinguishes the patches as the target or background. The appearance of a tracking target is usually irregular, however, the target bounding box is regular. As a result, the target bounding box contains some background regions. Since the patches are extracted from the bounding box, some of the patches may contain too much background regions, which may cause inaccurate tracking results. Moreover, the inaccurate tracking results may spread to the subsequent frames, thus leading to a drifting problem. Besides, the error produced in tracking process would gradually accumulate and propagate, resulting in poor performance in long-term tracking.

To combat this problem, the paper proposes a context adaptive learning method and for visual tracking. We propose a method to distinguish target and background regions from the bounding box. First, we formulate a weighted similarity function to construct a pure target region that has no background regions. The pure target region is used as a target prior and the surrounding area of a bounding box are used as a background prior. We exploit both the target prior and background prior to distinguish target and background regions from the bounding box. Patches which contain too much background regions could be discard to construct a robust appearance model of tracking targets.

2 The Proposed Context Adaptive Tracker

In this section, we will provide the details of the proposed context adaptive tracker. We use CNN-SVM [1] as baseline. We detail the process of extracting target region from target bounding box. This process is crucial to alleviate the drifting problem caused by inaccurate appearance representation of tracking target.

2.1 Superpixel-Based Pure Target Region Extraction

For a given image, the image domain consists of two parts: the target bounding box region T_r and its surrounding background region B_r. The region T_r could be obtained by the manually labeled ground truth $X = [x, y, w, h] \in \Re_4$ in the initial frame. (x, y) is the center location of the tracking target; w and h denote its weight and height in x-axis and y-axis, respectively.

At the initial frame of a video sequence, we segment the context region (The bounding box of the context region could be represented by $X_r = [x, y, \lambda w, \lambda h] \in \Re_4$. The $\lambda > 1$ is a constant parameter, which controls the size of the context region.) into N superpixels set $S = \{s_1, \cdots, s_n, \cdots, s_N\}$ for further processing. Here any edge preserving superpixel methods can be used and SLIC algorithm

is adopted in our paper. The input image is segmented into multiple uniform and compact region. For a certain superpixel S_N, we have:

$$l_n(S_n) = \begin{cases} 1, if & S_n \in T_p \\ 0, if & S_n \in B_r \end{cases} \tag{1}$$

where $l_n(S_n)$ denotes the label of superpixel s_n, the region T_p are set to $\gamma(\gamma < 1)$ time the target bounding box and is a constant parameter which is employed to construct a pure target region. The value of γ should not be set too large to ensure the region T_p to have abundant features of tracking target. However, tracking targets always have irregular sizes, leading to the region T_p still with some outliers which should be classified into background region. For further processing, it is worthy to construct a pure target region without outliers.

Given in the region T_p, we seek the most reliable target superpixels to construct a pure target region by using a weighted similarity function. Compared with the superpixels belonging to the pure target region, the number of outliers which are dissimilar with target in appearance, always occupy a little of proportion. The appearance of outliers always is different with the target superpixels, while several superpixels which are similar to a target superpixel can usually be found in the pure target region. Thus, a method that examines the similarity between each superpixel was proposed to detect outliers based on the Kernelized Correlation Filters.

$$c_i^p \doteq \sum_{S_j \in T_p} \max(\mathcal{F}(I(s_i)w_s^p)) \bigodot \mathcal{F}(I(s_j)w_s^p)w_{s_i}^p, s_i \in T_p \tag{2}$$

where c_i^p is defined as a likelihood of the superpixel s_i belonging to the tracking target or the background, $\mathcal{F}$ denotes the Fast Fourier Transform function and $\bigodot$ is the element-wise product, $I(s_i)$ denotes image intensity of a rectangular patch which could contain the superpixel s_i. A spatial weight function $w_{s_i}^p$ is defined by

$$w_{s_i}^p \doteq exp(-\frac{\|z_{s_i} - (x, y)\|^2}{(\sigma^p)^2}) \tag{3}$$

where context location z_{s_i} is the center of superpixel s_i, is a scale parameter of the region T_p.

The N_s number of patches with lowest score of c_i^p computed by Eq. (2) are regarded as outliers. N_s is an integer parameter which is used to control the number of outliers. Due to the outliers are more likely to be background, they should not be used to exploit target prior. A pure target region T_r^* is defined without the outliers. Assume that T_r^* is only composed by a set of superpixels which belong to tracking target, we only need to distinguish the rest of region $T_r^{'}$. And the $l_n(S_n)$ is updated as

$$l_n(S_n) = \begin{cases} 1, if & S_n \in T_r^* \\ 0, if & S_n \in B_r \end{cases} \tag{4}$$

2.2 Target and Background Likelihood

Assume pure target region as target prior and surrounding of the target bounding box as background prior, the similarity metrics between each superpixel $s_n \in T_r^{'}$, target and background likelihood are calculated for distinguishing the region $T_r^{'}$, respectively. Given a set of superpixels S, an undirected graph $\zeta = (\nu, \xi)$ is constructed to reveal the connection relationships (similarity metrics) between $T_r^{'}$, T_r^* and B_r, where $\nu = \{\nu_1, \ldots, \nu_n, \ldots, \nu_N\}$ denotes a set of nodes corresponding to superpixels set S and ξ is a set of undirected edges corresponding to nodes set ν. We define mean feature vectors ν_n^{lab} in the CIELAB color space and geometric center ν_n^{geo} on Euclidean distance for each node. The ν_n^{lab} and ν_n^{lab} are widely used in many algorithms to simplify further processing, due to their advantages of high-efficiency and superior appearance representation.

An initial regular graph where each node is only connected to its immediate neighbors is established by us. The adjacency matrix of the graph ζ is defined to be $A = [a_{ij}]_{N \times N}$. If the nodes ν_i and ν_j are immediate neighbors, then $a_{ij} \doteq 1$, otherwise $a_{ij} \doteq 0$. We subsequently define target and background edges based on the initial regular graph for calculating similarity metrics between $T_r^{'}$, T_r^* and B_r, respectively.

For each node $\nu_n \in T_r^*$, we consider it as an initial node, and connect it with its immediate and mediate neighbors. These neighbors are restricted to the target bounding box region T_r. The color feature constraint is considered to ensure the similarity between the initial node and its neighbors. Besides, the connections among these nodes are constrained by spatial geometric distance. The rule can be represented as:

$$\begin{aligned}
\xi_i^{T_r^*} \doteq & \{(s_i, s_j) | a_{ij} = 1, \nu_{ij}^{lab} \leq Th_{ij}^{T_r^*}, s_i \in T_r^*, s_j \in T_r\} \\
& \cup \{(s_i, s_n) | a_{jn} = 1, \nu_{in}^{lab} \leq Th_{in}^{T_r^*}, s_n \in T_r\} \\
& \cup \{(s_i, s_k) | a_{nk} = 1, \nu_{ik}^{lab} \leq Th_{ik}^{T_r^*}, s_k \in T_r\} \\
& \cup \{(s_i, s_l) | a_{kl} = 1, \nu_{il}^{lab} \leq Th_{il}^{T_r^*}, s_l \in T_r\} \\
Th_{ij}^{T_r^*} \doteq & \alpha_{T_r^*} \cdot \max \nu_{il}^{lab} \cdot \frac{w_n}{Z_j}, s_i \in T_r^*, s_j, s_m \in T_r
\end{aligned} \tag{5}$$

where the $\xi_i^{T_r^*}$ is a target edge corresponding to the node $\nu_i \in T_r^*$, $\nu_{ij}^{lab} \doteq \|\nu_i^{lab} - \nu_j^{lab}\|$ is a measure of visual similarity between superpixel s_i and s_j in the CIELAB color space, $\alpha_{T_r^*}$ is a fixed parameter which is used to ensure the similarity between the superpixel corresponding to the initial node and its neighbors, σ_j^{geo} is the variance of all the ν_j^{geo} corresponding to the set of superpixels $s^{geo} \doteq \{s_j | \nu_{ij}^{lab} \leq \alpha_{T_r^*} \cdot \max \nu_{im}^{lab}, s_i \in T_r^*, s_j \in T_r\}$, $\nu_{ij}^{geo} \doteq \|\nu_i^{geo} - \nu_j^{geo}\|$ is spatial geometric distance in euclidean space, $Th_{ij}^{T_r^*}$ is an adaptive threshold to determine whether the superpixel s_i is similar with the superpixel s_j, w_n is a threshold weight function and will be defined in detail in the following paragraphs, Z_j is a normalization factor to ensure that $\sum_{s_i \in s^{geo}} \frac{w_n}{Z_j} \doteq 1$.

Considering in the target bounding box region T_r, the central position of a superpixel which is near to the tracking target center, indicating that the superpixel is more likely to belong to tracking target. Furthermore, the near the geometric distance between the superpixel s_i and s_j is, the more likely that the superpixel s_j is similar to the superpixel s_i. Thus, different weight should be set for the adaptive threshold $Th_{ij}^{T_r^*}$ at different position of a superpixel. The threshold weight function is defined as:

$$w_n = \begin{cases} exp(-\dfrac{\nu_{ij}^{geo}}{(\sigma_{c_i^1}^{geo})^2}) & if \quad S_n \in T_r^* \\ exp(-\dfrac{\min \|\nu_n^{geo} - \nu_{center}^{geo} \pm (\frac{w}{2\gamma}, \frac{h}{2\gamma})\|^2}{(\sigma^{geo})^2}) & \\ \times exp(-\dfrac{\nu_{ij}^{geo}}{(\sigma_{c_i^1}^{geo})^2}) & if \quad S_n \in T_r^{'} \end{cases} \tag{6}$$

where σ^{geo} is the variance of all the ν_n^{geo} corresponding to the set of superpixels $s_n \in T_r^{'}$, ν_{center}^{geo} is geometric center of the superpixel which locates to the center of the tracking target, $exp(-\frac{\nu_{ij}^{geo}}{(\sigma_{c_i^1}^{geo})^2})$ measures the spatial variance between the superpixel s_i and s_j. Given a superpixel $s_n \in T_r^{'}$, the larger the value of the threshold weight function, the more probable it will be connected with the superpixel $s_i \in T_r^*$.

We could obtain the each target edge $\xi_i^{T_r^*}$ through Eq. (5) and define the set of target edges $\xi^{T_r^*}$ as $\xi^{T_r^*} \doteq [\xi_1^{T_r^*} \cdots \xi_i^{T_r^*} \cdots \xi_{N_T}^{T_r^*}]$, where N_T is the total number of superpixels which belong to the pure target region T_r^*. In each target edge $\xi_i^{T_r^*}$, all the superpixels are similar with the superpixel s_i which belongs to the T_r^*. If a superpixel $s_n \in T_r^{'}$ is discovered multiple times in the target edges $\xi^{T_r^*}$, it shows that the superpixel is actually quite similar with the pure target region T_r^* and should be categorized into tracking target region. Then we define the target similarity metric Sim_n^T as:

$$Sim_n^T \doteq \frac{N_n^T}{N_T}, \{n | s_n \in T_r^{'}\} \tag{7}$$

where the N_n^T is the number of times that the superpixel $s_n \in T_r^{'}$ appears in the $\xi^{T_r^*}$.

Besides target prior, we also exploit background prior for distinguishing the region $T_r^{'}$. For a robust performance of visual tracking, our purpose is to distinguish the background region from the target bounding box, which may lead to drift problem. The rule can be represented as:

$$\begin{aligned}\xi_i^{B_r} \doteq \{(s_i, s_j)|a_{ij} = 1, \nu_{ij}^{lab} \leq \alpha_{B_r} \cdot \max \nu_{in}^{lab} \\ s_i \in B_r, s_j, s_n \in B_r \cup T_r^{'}\} \\ \cup\{(s_i, s_k)|a_{jk} = 1, \nu_{ik}^{lab} \leq \alpha_{B_r} \cdot \max \nu_{in}^{lab} \\ s_n, s_k \in B_r \cup T_r^{'}\}\end{aligned} \tag{8}$$

where the edge ξ^{B_r} is a background edge corresponding to the node $v_i \in B_r$, α_{Br} is a fixed parameter which promise the visual similarity between superpixel s_i, s_j and s_n in the CIELAB color space.

Similarly, the set of background edges ξ^{B_r} is defined as $\xi^{B_r} = [\xi_1^{B_r} \ldots \xi_i^{B_r} \ldots \xi_{N_B}^{B_r}]$, where N_B is the total number of superpixels which belong to the surrounding background region of target bounding box. Then we define the background similarity metric Sim_n^Bas:

$$Sim_n^B \doteq \frac{N_n^B}{N_B}, \{n|s_n \in T_r^{'}\} \tag{9}$$

where the N_n^B is the number of times that the superpixel $S_n \in T_r^{'}$ appears in the ξ^{B_r}.

2.3 Classification for Superpixel

As aforementioned, we calculate the similarity metrics between each superpixel $S_n \in T_r^{'}$, target and background, respectively. Based on these, we could classify the each superpixel $S_n \in T_r^{'}$, to indicate whether it belongs to the target or the background. The label of the superpixel is determined by

$$l_n(S_n) = \begin{cases} 1, if & Sim_T \geq Sim_B \\ 0, if & Sim_T < Sim_B \end{cases} \tag{10}$$

Through Eqs. (4) and (10), we could obtain a complete region of tracking target.

3 Experiments

In this section, extensively experiments are conducted for the proposed CAVT method. We first introduce the details of our experimental setup including parameters, dataset, and evaluation metrics. We then evaluate our method on an online object tracking benchmark with comparisons to state-of-the-art methods.

3.1 Experimental Setup

The constant parameter is set to 1.35, which means that the size of context region is initially set to 1.35 times the size of the ground truth target bounding box. The SLIC algorithm [2] is applied to extract superpixels from the context

region where the maximal number of superpixels is set 200. The value of outliers are fixed at 0.8 and 6, respectively, which are used to make sure a pnrificatus target regions. The threshold α_{Br} and $\alpha_{T_r^*}$ are empirically defined as 0.1 and 0.15, respectively. We note that the parameters are fixed in all the experiments. As for the rest of parameters, we use the default setting of the base CNN-SVM tracker to prove that our method could improve the effectiveness for part-based trackers. Dataset and Evaluation Metrics:

We report the evaluation of our proposed CAVT method on the CVPR2013 Online Object Tracking Benchmark (OOTB) [3] that contains 50 fully annotated sequences with comparisons to state-of-the-art methods. The OOTB is a comprehensive benchmark specifically designed for evaluating tracking performance, which extensively used in the online tracking literature over the past several years. In OOTB, the quantitative evaluation for the effectiveness of different trackers is based on four types of metrics. The first metric is mean Center Location Error (CLE), which is defined as the average Euclidean distance between the center of tracking result and the ground truth for each frame. The second metric is Pascal VOC Overlap Ratio (VOR), which is defined as $VOR \doteq \frac{Area(B_T \bigcap B_G)}{Area(B_T \bigcup B_G)}$, where B_G and B_T denote the bounding box of ground truth and the tracking results, respectively. The rest of metrics are precision plot and success plot which can measure the overall performance of the different trackers. The precision plot demonstrates the percentage of successfully tracked frames on which the CLE of a tracker is within a given threshold. The success plot also illustrates the percentage of successfully tracked frames by measuring the Intersection Over Union (IOU) metrics on each frame. The area under curve (AUC) score is used to rank the tracking algorithms in both the precision plot and success plot.

3.2 Comparison with State-of-the-Arts

In this paper, we compare the proposed tracker on the OOTB with 34 representative tracking methods. Among the competitor trackers, we first consider those 29 popular approaches whose results are available in OOTB including TLD [4], etc. The 29 popular trackers can be referred to [3] in details. And on top of these, other 6 recently published state-of-the-art trackers with their shared source code: CNN-SVM [1], RPT [5], KCF [6], CNT [7], SAMF [8] and MEEM [9].

To quantitatively compare all the 34 trackers, we use the original software provided by [3] to compute both precision and success plots in Fig. 1. Following the setting in [3], we conduct all the experiments using one-pass evaluation (OPE) strategy for fair comparison with the state-of-the-art trackers. The OPE is computed by running a tracker throughout a video sequence with initialization by the ground truth in the initial frame. The performance gap between our tracker and the second best tracker in the literature is 0.3% in tracking precision measure and 2.7% in success measure under OPE; the proposed tracker achieves 85.5% and 62.4% accuracy while the base tracker is 85.2% and 59.7% (CNN-SVM). It is obvious that both the precision and success plots demonstrate that the proposed tracker performs well against the competitors. The precision and success plots

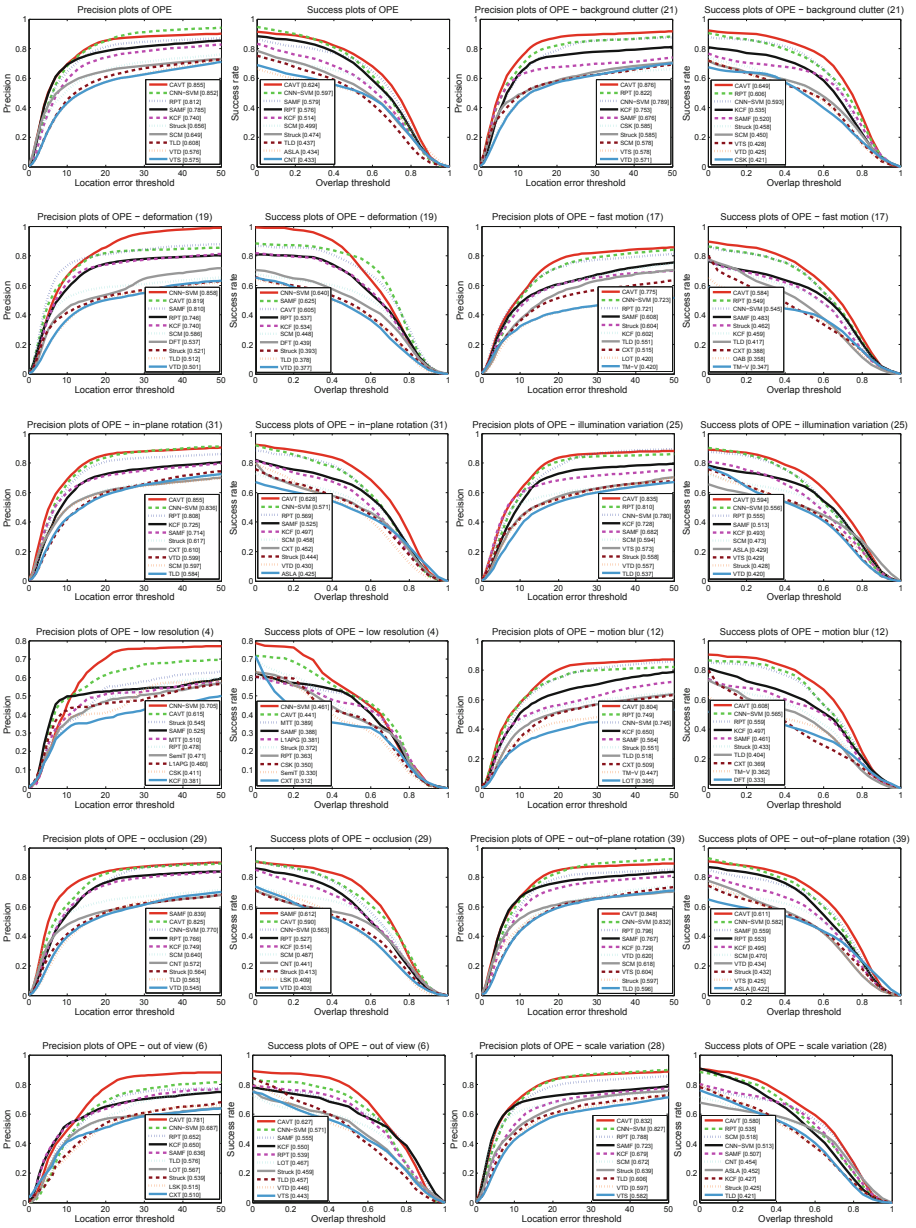

Fig. 1. Precision and success plots of the OPE and Overlap success plots for 11 challenging attributes. The legend contains the AUC score for each tracker. The proposed CAVT method performs favorably against the state-of-the-art trackers when evaluating with 11 challenging factors.

illustrate the overall performance over all the 50 sequences. For better evaluation and analysis of the strength and weakness of tracking approaches, we analyze the performance of trackers based on the 11 attributes of image sequences in Fig. 1. Note that, the proposed CAVT performs well, especially in dealing with challenging factors including BC, FM and IPR. For each plot, only the top ten trackers are displayed for presentation clarity.

4 Conclusion

In this paper, we propose a generic context adaptive learning approach for improving the performance of CNN-based methods which use the sliding-window method to generate candidate samples. To overcome the drifting problem of state-of-the-art CNN-based trackers, we exploit the intrinsic relationship among target regions and background regions to identify distracting regions containing too much background. Extensive experiment results on benchmark dataset demonstrates that the proposed CAVT method can achieve competitive accuracy on challenging sequences and significantly improve the performance of the base tracker.

References

1. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: International Conference on International Conference on Machine Learning (2015)
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
3. Yi, W., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Computer Vision and Pattern Recognition (2013)
4. Zdenek, K., Krystian, M., Jiri, M.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
5. Yang, L., Zhu, J., Hoi, S.C.H.: Reliable patch trackers: robust visual tracking by exploiting reliable patches. In: Computer Vision and Pattern Recognition (2015)
6. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
7. Zhang, K., Liu, Q., Wu, Y., Yang, M.-H.: Robust visual tracking via convolutional networks without training. IEEE Trans. Image Process. **25**(4), 1779–1792 (2016)
8. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8926, pp. 254–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_18
9. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 188–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_13