ABSTRACT

Recently, some correlation filter based trackers with detection proposals have achieved state-of-the-art tracking results. However, a large number of redundant proposals given by the proposal generator may degrade the performance and speed of these trackers. In this paper, we propose an adaptive proposal selection algorithm which can generate a small number of high-quality proposals to handle the problem of scale variations for visual object tracking. Specifically, we firstly utilize the color histograms in the HSV color space to represent the instances (i.e., the initial target in the first frame and the predicted target in the previous frame) and proposals. Then, an adaptive strategy based on the color similarity is formulated to select high-quality proposals. We further integrate the proposed adaptive proposal selection algorithm with coarse-to-fine deep features to validate the generalization and efficiency of the proposed tracker. Experiments on two benchmark datasets demonstrate that the proposed algorithm performs favorably against several state-of-the-art trackers.

Index Terms— Visual Tracking, Correlation Filters, Detection Proposals, Convolutional Neural Network

1. INTRODUCTION

Visual object tracking plays an important role in many applications of computer vision, such as video surveillance, human-computer interface and robotic analysis. One of the main challenges of object tracking is to handle the scale variations of targets caused by deformation, fast motion and rotation, etc.

In recent years, the correlation filter based trackers [1, 2] have attracted much attention due to their high efficiency and accuracy. These trackers mainly learn the correlation filters in the Fourier domain to detect the most likely candidate in each frame. In particular, the convolution operation with the correlation filters in the spatial domain corresponds to the element-wise operation in the Fourier domain, leading to the high efficiency of object tracking.

There are two main ways to address the scale variations of targets during tracking. One common way is to design several scales empirically or employ the extra correlation filters to select the best scale for each frame. For example, the HCF tracker [3] utilizes three layers from the deep network as three separate models. Then, it extracts the HOG features from the patches with different scales to decide the best scale for each frame. The SAMF tracker [4] copes with the scale variations by designing some scales and using the correlation filters with different scales to obtain the final scale. DSST [5] and fDSST [6] formulate a correlation filter based on the DCF tracker [1] and another one-dimensional correlation filter to estimate the location and scale, respectively. These algorithms are based on the fixed aspect ratio of the ground-truth in the initial frame, and scales are empirically designed, thus resulting in sub-optimal tracking accuracy.

The second way is to combine the detection proposal generators with correlation filters to handle scale variations [7, 8]. For instance, KCFDPT [9] utilizes KCF [1] to detect the initial location in each frame, where the EdgeBoxes algorithm [10] is adopted to generate proposals. Owing to the effective combination of detection proposals and correlation filters, KCFDPT achieves more accurate tracking performance than fDSST and SAMF. However, the redundant proposals generated by EdgeBoxes hinder the efficiency of KCFDPT. Moreover, the distractors in the redundant proposals generated by the edge information may degrade the tracking performance in consecutive frames.

To solve the above problems, we propose an algorithm to adaptively select a small number of high-quality proposals and effectively handle the scale variations for correlation filter based trackers. Our contributions are summarized as follows:

- We employ the color information to measure the similarity between the instances and proposals generated by EdgeBoxes. Based on the HSV color histogram, we propose an adaptive selection strategy to discard the redundant proposals based on the confidence of the current correlation filter. Therefore, we decrease the distractors in the redundant proposals and improve the tracker with better performance and faster speed.

- We further integrate the proposal selection algorithm
with coarse-to-fine deep features derived from the VG- 
GNet [11] to demonstrate the high quality of the se-
lected proposals. Extensive experimental results on two
benchmarks (OTB2013 [12], OTB2015 [13]) demon-
strate that the adaptive proposal selection algorithm ef-
effectively improves the tracking performance, espe-
ically in terms of scale variations.

2. BASELINE ALGORITHM

In this section, we present a brief introduction of the
KCFDPT [9] algorithm, which is the baseline tracker in this
paper. KCFDPT contains two parts: the KCF tracker [1] and
the proposal generator that uses EdgeBoxes [10] with back-
ground suppression.

In KCF, the objective of the correlation filter formulation
is to learn a correlation filter $\omega$ and minimize the squared er-
ror over a set of samples $\{x_1, x_2, \ldots, x_j, \ldots, x_n\}$ and the cor-
responding regression targets $\{y_1, y_2, \ldots, y_j, \ldots, y_n\}$, where $j$
ranges from 1 to $n$. $x_j$ is the $j$-th cyclic shift of the base sam-
ple $x_1$. $y_j$ is the corresponding label generated by a Gaussian
function. $y_j$ is the label for the base sample $x_1$, which equals
to 1. The problem can be written as:

$$
\min_\omega \sum_j^n (f(x_j) - y_j)^2 + \lambda \|\omega\|^2,
$$

(1)

where $\lambda$ denotes a regularization parameter to alleviate the
over-fitting problem. $n$ denotes the number of the training
samples. Owing to the properties of the circulant matrix, the problem of Eq. (1) has the closed-form solution, which is:

$$
\hat{\alpha} = \frac{\hat{y}}{k^{xz:x_1} + \lambda},
$$

(2)

where $\alpha$ represents the parameter matrix of the correlation
filter in dual space, as opposed to $\omega$ in the primal space. The
hat means the discrete Fourier transform. $k^{xz:x_1}$ represents the
kernel correlation operation of the base sample $x_1$.

Given an image patch $z$, KCF will be utilized to obtain
the response map and detect the location of the target. The
response map in KCF is computed as follows:

$$
\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha},
$$

(3)

where $\hat{f}(z)$ is the response map of the image patch $z$ in the
Fourier domain, whose maximum in the spatial domain indi-
cates the detection location and confidence. $\hat{k}^{xz}$ denotes the
kernel correlation operation between the current appearance
model $\bar{x}$ and the image patch $\hat{z}$ in the Fourier domain. $\odot$ de-
notes the element-wise product.

KCFDPT utilizes EdgeBoxes [10] to assign the back-
ground suppression weights to edges intersecting the bound-
ary of the image patch and calculate the edge response value
$r_i$ of each pixel $i$. After obtaining the response value of each
pixel from the background suppression factors, the score for
a bounding box $b$ is evaluated by:

$$
h_b = \frac{\sum_{i \in b} c_i r_i}{2(b_u + b_v)^\kappa} - \frac{\sum_{i \in b^m} r_i}{2(b_u + b_v)^\kappa},
$$

(4)

where $r_i$ denotes the edge response value of a pixel $i$ within
a bounding box $b$. $b_u$ and $b_v$ are the width and height of $b$, re-
spectively. $b^m$ stands for the central region of $b$, whose size
is $b_u/2 \times b_v/2$. $c_i \in [0, 1]$ is a parameter, measuring how
likely the contour that $i$ belongs to, is wholly contained in $b$.
$\kappa$ is also a parameter to penalize the boxes with the large size.

In the detection stage, KCFDPT employs KCF to local-
ize the center of the detection patch. Then, KCFDPT utilizes the parameters (e.g., intersection over union (IoU)) to select
proposals generated by EdgeBoxes with the background sup-
pression. The response map of each proposal can be obtained
by Eq. (3), and the proposal with the highest response will
be selected as the most promising proposal. Finally, the most
promising proposal will be used to localize the target and up-
date the correlation filter model $\alpha$ in Eq. (2).

Note that the proposals in KCFDPT are generated by using
the EdgeBoxes algorithm, which only considers the edge
information in the detection patch. Therefore, the proposals
are not robust enough to handle motion blur and scale vari-
ations. Furthermore, the proposals contain the redundant in-
formation that may degrade the tracking performance. In this
paper, we propose to exploit the color information to adaptively
select a small number of high-quality proposals to improve and
accelerate the baseline KCFDPT tracker.

3. OUR ALGORITHM

The pipeline of the correlation filter based tracker with
the adaptive proposal selection is shown in Fig. 1. During the
tracking process, we firstly utilize the correlation filter based
tracker to obtain the initial location and the target size for the
current frame, as shown in Fig. 1(b). Then, EdgeBoxes with
background suppression is adopted to generate detection pro-
posals, which are further selected by the IoU constraint, as
shown in Fig. 1(c). Next, based on the HSV color histograms
(Section 3.1) of both the proposals and instances, the adap-
tive proposal selection (Section 3.2) is utilized to discard the
redundant proposals, where the correlation filter model will
select the proposal with the highest response, as shown in
Fig. 1(d). Finally, the prediction for the current frame is a
trade-off between the most promising proposal and the initial
prediction given by KCF, as shown in Fig. 1(e). We further
integrate the proposal selection algorithm into the deep fea-
tures based correlation filters (Section 3.3) for robust object
tracking.
3.1. Color similarity measurement

When the $i$-th frame comes, the location $o_{i-1}$ and target size $(w_{i-1}, h_{i-1})$ from the previous frame will be used as the inputs of KCF to detect the initial predicted location $o'_i$. We keep the instance $I_{i-1}$ from the previous frame with the location $o'_{i-1}$ and size $(w_{i-1}, h_{i-1})$. In this paper, the initial instance $I_1$ (i.e., the initial target in the first frame) with size $(w_1, h_1)$ in the first frame and the previous instance $I_{i-1}$ (i.e., the predicted target in the previous frame) are two instances that we maintain during the tracking process. The EdgeBoxes with background suppression algorithm is performed on a detection window patch $z_d$. The center location and size of $z_d$ are $o'_i$ and $(s_d w_{i-1}, s_d h_{i-1})$. $s_d$ is a parameter to render the detection window slightly larger than the previous size $(w_{i-1}, h_{i-1})$. The output of EdgeBoxes with background suppression contains the redundant proposals, as visualized in Fig. 1(c). Based on these proposals, we further select the proposals which are similar to these two instances, namely, $I_{i-1}$ in the previous frame and $I_1$ in the first frame.

HSV color space is shown to have better results for image retrieval than RGB color space [14]. The proposals and two instances are represented in the HSV color space. More specifically, for each pixel, we firstly normalize the three channels of HSV (i.e., hue, saturation, and value) into the range of $[0, 255]$. Secondly, we synthesize the image of three channels into the image of one channel, whose values range from 0 to 255. Therefore, we uniformly quantify the values of the three channels into 16, 4, 4 levels, respectively. Then, we multiply the three values of each pixel by 16, 4, 1 and add them together. As a result, each pixel can be represented by 8 bits, where the first 4, middle 2 and last 2 bits stand for the 16, 4, 4 levels of hue, saturation, and value, respectively.

Mathematically, the color histogram of each proposal or instance in the HSV color space can be obtained by:

$$H = Hist(16 \times Q_{16}(P_{hsv}(m, n, 1)) + 4 \times Q_{16}(P_{hsv}(m, n, 2)) + Q_{16}(P_{hsv}(m, n, 3))),$$

where $Q_{16}(P_{hsv}(m, n, 1))$ represents that the pixel value in the $m$-th row and the $n$-th column in the first channel of the proposal $P$ is quantified into 16 levels. $Q_{4}(P_{hsv}(m, n, 2))$ represents that the pixel value in the second channel of the proposal $P$ is quantified into 4 levels. $Q_{4}(P_{hsv}(m, n, 3))$ represents the similar meaning for the third channel. $Hist$ stands for the counting procedure of generating the color histogram, which ranges from 0 to 255. The histogram of instance $I$ can be calculated similar to the proposal $P$.

After obtaining the color histograms of two instances and the proposals in the current frame, we further employ the Bhattacharyya coefficient to measure the similarity in the HSV color space. The similarity between the instance $I$ ($I_1$ or $I_{i-1}$) and the proposal $P$ can be computed as follows,

$$Sim(I, P) = \sum_{r \in R} \frac{H_I(r) H_P(r)}{N(I) N(P)},$$

where $R$ is the range of bins in the color histogram, which is $[0, 255]$. $H_I(r), H_P(r)$ represent the color histogram vectors of the instance $I$ and the proposal $P$, respectively. $N(I), N(P)$ denote the number of pixels in the instance $I$ and the proposal $P$, respectively. In this way, the fraction denotes the normalization operation of color histogram.

3.2. Adaptive proposal selection

The instance $I_{i-1}$ may be contaminated during the tracking process, so we propose an adaptive proposal selection strategy to choose the informative proposals. For each frame, we update the mean confidence of correlation filters as,

$$f_{i}^{\text{mean}} = (1 - \eta) f_{i-1}^{\text{mean}} + \eta f_{i}^{\text{max}},$$

where $f_{i-1}^{\text{mean}}$ is the mean confidence of correlation models from the previous frames. $f_{i}^{\text{mean}}$ is the corresponding confidence from the first frame to the current frame. $f_{i}^{\text{max}}$ denotes the maximum value of the response map of the final selected proposal in the $i$-th frame. $\eta$ is a model confidence factor.

For the $i$-th frame, KCF is employed to obtain the response map and localize the center of detection window. We compare the temporary maximum response value $f_{i}^{\text{max}}$ from KCF with $f_{i-1}^{\text{mean}}$ to indicate whether the instance $I_{i-1}$ is reliable or not. When $f_{i}^{\text{max}} < \eta' f_{i-1}^{\text{mean}}$, it indicates the instance $I_{i-1}$ is likely to be contaminated and unreliable. $\eta'$ is a rate.
to find those instances with too small response values. Therefore, we count the number of contaminated frames $\Delta_i$ as,

$$
\Delta_i = \sum_{t=i_1}^{t=i_2} \frac{1}{2} (\text{sign}(\eta' f_{\text{mean}}^{t-1} - f_{\text{max}}^t) + 1),
$$

where $\text{sign}$ is a sign function and $i_1$ denotes the frame number of the previous confident instance. The final score of the proposal $P$ can be calculated by:

$$
S_P = (1 - e^{(-\alpha D \Delta_i)}) \text{Sim}_P^{t_1} + e^{(-\alpha D \Delta_i)} \text{Sim}_P^{t_{i-1}},
$$

where $\Delta_i$ is the number of the contaminated frames and $\alpha D$ is a trade-off parameter. $\text{Sim}_P^{t_1}$ and $\text{Sim}_P^{t_{i-1}}$ denote the color scores between the proposal $P$ and the instances $I_i$ and $I_{i-1}$, respectively. When $\Delta_i$ is larger, it indicates that the previous instance $I_{i-1}$ is not reliable enough to select proposals (the instance $I_{i-1}$ is contaminated with the high probability) and we should mainly rely on the uncontaminated instance $I_i$.

Based on Eq. (9), we rank the proposals in the descending order according to the similarity between each proposal and two instances. Then we discard about half of proposals to remove the distractors and accelerate the tracker. The results of proposal selection can be visualized in Fig. 1(d). After we obtain these selected proposals, Eq. (3) is used to obtain the response map of each high-quality proposal. The proposal with the highest response is the candidate proposal. From Fig. 1, we can see that the proposal selection algorithm is effective for the correlation filter based trackers.

To avoid the over-sensitive problem and reduce the estimation error, a damping factor $\beta$ is used to obtain the target state in the $i$-th frame and keep a balance between the initial prediction and proposal selection. Assume that the proposal $P$ with the location $o_i^P$ and the size $(w_i^P, h_i^P)$ is the final selected proposal. The target state fine-tuning process is formulated as follows,

$$
o_i = o_i' + \beta(o_i^P - o_i'),
$$

$$
(w_i, h_i) = (w_{i-1}, h_{i-1}) + \beta((w_i^P, h_i^P) - (w_{i-1}, h_{i-1})),
$$

where $o_i'$ and $(w_{i-1}, h_{i-1})$ denote the initial location by KCF and the previous size, $o_i$ and $(w_i, h_i)$ are the final prediction in the $i$-th frame. The prediction will be used to localize the target and update the correlation filter model $\alpha$ in Eq. (2).

### 3.3. Integrating with deep features

The baseline KCFDPT tracker [9] integrates KCF with color naming, image intensity and HOG features. Since the proposed adaptive proposal selection algorithm is generic and can be combined with different correlation filter based trackers, we also integrate the coarse-to-fine deep features (as used in the HCF tracker [3]) with the proposed algorithm to verify the generalization of the proposed algorithm.

More specifically, when the $i$-th frame comes, the location $o_{i-1}$ and target size $(w_{i-1}, h_{i-1})$ from the previous frame are used to extract different layers of features (i.e., coarse-to-fine features) from VGG-Net. Deep features contain more semantic information than shallow features, but the resolutions of the deep features and shallow features are different. Hence, the bilinear interpolation operation is utilized to ensure that the shallow and deep features can be fused together. Then, the response map of deep features can be obtained by,

$$
g(z) = \sum_{d=1}^{D} \mu_d F^{-1}(\hat{\omega}_d \odot z_d^*),
$$

where $D$ is the number of layers of deep features we use. $\mu_d$ denotes the weight of the $d$-th layer and $F^{-1}$ denotes the inverse discrete Fourier function. $\odot$ denotes the element-wise product. $\hat{\omega}_d$ stands for the $d$-th correlation filter model matrix in the Fourier domain. $z_d^*$ represents the current feature of the patch $z$ in the $d$-th layer in the Fourier domain. And $*$ denotes the complex-conjugate operation. The maximum of $g(z)$ indicates the initial location in the current frame.

After obtaining the initial location by Eq. (11), the Edge-Boxes algorithm with background suppression will be employed to generate some proposals (Section 2). Based on these proposals, half of them will be discarded according to the similarity between instances and proposals (Section 3.1.1 and 3.1.2). Finally, the proposals will be evaluated by Eq. (11) to find the proposal with the highest response, and the final prediction in the current frame is determined by Eq. (10) from the candidate proposal and the initial prediction.

### 4. EXPERIMENTS

We perform comprehensive experiments on two benchmarks: OTB2013 [12] and OTB2015 [13]. And we also evaluate the tracking performance of different trackers under the scale variation attribute.

#### 4.1. Implementation details and parameter settings

To show the effectiveness of the proposed proposal selection algorithm, we implement the proposed CFAPS tracker (Correlation Filter tracking with Adaptive Proposal Selection) based on the original KCFDPT, where the hand-crafted features are used. That is, we concatenate color naming, image intensity and HOG features directly in CFAPS. In addition, we incorporate the adaptive proposal selection algorithm into the HCF tracker [3]. The tracker is named as DeepCFAPS. In the DeepCFAPS tracker, we utilize the outputs of the conv3-4, conv4-4 and conv5-4 convolutional layers from the VGG-Net-19 [11] as features. The values of $\mu_d$ of each layer in Eq. (11) are respectively set to 0.25, 0.50 and 1.0, which are similar to [3].

The regularization parameter $\lambda$ in Eq. (1) and Eq. (2) is set to $10^{-4}$. For the proposal generator, the parameter $s_d$ in
Table 1: Analysis of selecting different percentages of proposals on the tracking performance of the OTB2015 dataset. Note that $I_1$ or $I_{i-1}$ denotes that only one instance $I_1$ or $I_{i-1}$ is used for similarity measurement during tracking.

| Percentage(%) | 30  | 50  | 50 ($I_1$) | 50 ($I_{i-1}$) | 70  | 100 |
|---------------|-----|-----|------------|---------------|-----|-----|
| DP(%)         | 74.3| 77.2| 73.8       | 74.4          | 76.2| 74.7|
| AUC(%)        | 54.1| 56.4| 54.0       | 54.2          | 55.4| 54.8|

the detection window size is set to 1.40. The scale penalty parameter $\kappa$ in Eq. (4) is set to 1.40. The damping factor $\beta$ in Eq. (10) is set to 0.70. The above parameters are totally the same as those in KCFDPT. The confidence factor $\eta$ in Eq. (7) and the rate $\eta'$ in Eq. (8) are set to 0.01 and 0.60, respectively. The parameter $\alpha_D$ in Eq. (9) is set to 0.15.

For evaluation metrics, DP (the distance precision at 20 pixels threshold in precision from one pass evaluation (OPE)), and AUC (the area under curve in success plot for OPE) are used in this paper. The OTB 2013 and OTB 2015 datasets are annotated with 11 attributes, including illumination, scale variation, occlusion, deformation, etc. Specifically, We use DP of scale variation to evaluate the tracking performance under the scale variation attribute.

In Table 1, we analyze the impact of selecting different percentages of proposals on the performance of CFAPS on the OTB 2015 dataset. Selecting 100% of proposals corresponds to the KCFDPT tracker. We can see that CFAPS with half of the proposals can achieve the best performance. The results obtained by selecting about 70% and 100% of proposals are slightly inferior to those obtained by selecting about 50% of proposals, which is mainly caused by the distractors contained in the redundant proposals. Moreover, we also show the results when only using the initial instance or the previous instance for the proposal selection. The proposed CFAPS tracker can achieve the best performance, which demonstrates the importance of adaptive selection strategy. In the following sections, we will select about 50% of proposals.

4.2. Comparison with the state-of-the-art algorithms

We respectively evaluate the performance of CFAPS and DeepCFAPS compared with state-of-the-art algorithms.

Evaluation of CFAPS. To show that the proposed adaptive proposal selection algorithm is effective for selecting high-quality proposals to handle scale variations, in this subsection, we compare the CFAPS tracker with several state-of-the-art trackers which are mainly designed for scale variations: SAMF [4], DSST [5], iDSST [6], KCFDPT [9], KCFDPT [8] and KCF [1]. All experiments are conducted on the Intel i7 3.6GHz CPU. The comparison results are given in Fig. 2.

CFAPS achieves the top performance among these trackers. Compared with KCF, our tracker outperforms it by a large margin on the DP metric (i.e., 9.4%/9.2% on OTB2013 and OTB2015, respectively). On the OTB2013 dataset, CFAPS improves KCFDPT by 2.0%, 2.2% and 1.1% on the DP, DP of scale variation and AUC metrics. On the OTB2015 dataset, CFAPS outperforms KCFDPT by 2.5%, 1.9% and 1.6% on the DP, DP of scale variation and AUC metrics, respectively. Furthermore, KCFDPT runs at 28.6 frames per second (fps), but CFAPS can run at 40.1 fps on average on the OTB2015 dataset, which shows the proposed adaptive proposal selection algorithm can improve the efficiency of the KCFDPT tracker. In general, CFAPS achieves better performance than KCFDPT in terms of tracking accuracy and speed.

Evaluation of DeepCFAPS. We compare the proposed DeepCFAPS with DeepKCFDPT (integrating KCFDPT with HCF [3]) and several state-of-the-art trackers: SAMF [4], DSST [5], iDSST [6], KCFDPT [9], KCFDPT [8] and KCF [1]. All experiments are conducted on the NVIDIA GTX TITAN GPU. The comparison results are given in Fig. 3. Note that the HCF tracker in our experiments deploys an extra scale scheme.

As shown in Fig. 3, DeepKCFDPT does not outperform HCF with the scale scheme on the DP and AUC metrics. On the OTB2015 dataset, CFAPS outperforms KCFDPT by 2.5%, 1.9% and 1.6% on the DP, DP of scale variation and AUC metrics, respectively. Furthermore, KCFDPT runs at 28.6 frames per second (fps), but CFAPS can run at 40.1 fps on average on the OTB2015 dataset, which shows the proposed adaptive proposal selection algorithm can improve the efficiency of the KCFDPT tracker. In general, CFAPS achieves better performance than KCFDPT in terms of tracking accuracy and speed.
large margin of 5.9%. On the OTB2015 dataset, our tracker is not as good as DeepSRDCF on the DP and AUC metrics, but it still outperforms the HCF tracker. Especially, our tracker outperforms DeepSRDCF and HCF by 1.9% and 3.5% on the DP of scale variation metric.

Fig. 4 shows some tracking results obtained by three trackers: KCFDPT, CFAPS and DeepCFAPS. When scale variations are caused by fast motion and motion blur (e.g., Soccer), the KCFDPT tracker chooses those distractors and focuses on the local part of the target. Moreover, when scale variations caused by large deformation occur (e.g., Gym), the bounding boxes of KCFDPT cannot adapt to the appearance of target. Especially, the tracker KCFDPT drifts when the scale variation occurs in Human9. In contrast, the proposed CFAPS and DeepCFAPS trackers can track these targets well.

5. CONCLUSION

In this paper, we propose an adaptive proposal selection algorithm for object tracking to effectively handle scale variations. We integrate the proposal selection algorithm with both hand-crafted and deep features to verify the generalization and effectiveness of the algorithm. Extensive experiments on two challenging datasets demonstrate the superiority of the proposed trackers against several state-of-the-art trackers, especially in terms of the scale variations.

6. REFERENCES

[1] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, “High-speed tracking with kernelized correlation filters,” IEEE TPAMI, vol. 37, no. 3, pp. 583–596, 2015.
[2] Ruize Han, Qing Guo, and Wei Feng, “Content-related spatial regularization for visual object tracking,” in ICME, 2018.
[3] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, “Hierarchical convolutional features for visual tracking,” in ICCV, 2015.
[4] Yang Li and Jianke Zhu, “A scale adaptive kernel correlation filter tracker with feature integration,” in ECCVW, 2014.
[5] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg, “Accurate scale estimation for robust visual tracking,” in BMVC, 2014.
[6] Martin Danelljan, Gustav Häger, Fahad Shahtz Khan, and Michael Felsberg, “Discriminative scale space tracking,” IEEE TPAMI, vol. 39, no. 8, pp. 1561–1575, 2017.
[7] Yanjie Liang, Qiangqiang Wu, Yi Liu, Yan Yan, and Hanzi Wang, “Robust correlation filter tracking with shepherded instance-aware proposals,” in ACM MM, 2018.
[8] Dafei Huang, Lei Luo, Mei Wen, Zhaoyun Chen, and Chunyuan Zhang, “Enable scale and aspect ratio adaptability in visual tracking with detection proposals,” in BMVC, 2015.
[9] Dafei Huang, Lei Luo, Zhaoyun Chen, Mei Wen, and Chunyuan Zhang, “Applying detection proposals to visual tracking for scale and aspect ratio adaptability,” IJCV, vol. 122, no. 3, pp. 524–541, 2017.
[10] C Lawrence Zitnick and Piotr Dollár, “Edge boxes: Locating object proposals from edges,” in ECCV, 2014.
[11] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.
[12] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Online object tracking: A benchmark,” in CVPR, 2013.
[13] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Object tracking benchmark,” TPAMI, vol. 37, no. 9, pp. 1834–1848, 2015.
[14] Shamik Sural, Gang Qian, and Sakti Pramanik, “Segmentation and histogram generation using the hsv color space for image retrieval,” in ICIP, 2002.
[15] Martin Danelljan, Gustav Häger, Fahad Shahtz Khan, and Michael Felsberg, “Convolutional features for correlation filter based visual tracking,” in ICCVW, 2015.