EACOFT: an energy-aware correlation filter for visual tracking.

LIU, Q., REN, J., WANG, Y., WU, Y., SUN, H. and ZHAO, H.

2021

EACOFT: An energy-aware correlation filter for visual tracking

Qiaoyuan Liu[a], Jinchang Ren[b,c,*], Yuru Wang[d], Yuanbo Wu[d], Haijiang Sun[a], Huimin Zhao[b]

[a] *Changchun Institute of Optics,Fine Mechanics and Physics, CAS, Changchun, China*
[b] *School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China*
[c] *Dept of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK*
[d] *School of Information science and technology, Northeast Normal University, Changchun, China*

Abstract

Correlation filter based trackers attribute to its calculation in the frequency domain can efficiently locate targets in a relatively fast speed. This characteristic however also limits its generalization in some specific scenarios. The reasons that they still fail to achieve superior performance to state-of-the-art (SOTA) trackers are possibly due to two main aspects. The first is that while tracking the objects whose energy is lower than the background, the tracker may occur drift or even lose the target. The second is that the biased samples may be inevitably selected for model training, which can easily lead to inaccurate tracking. To tackle these shortcomings, a novel energy-aware correlation filter (EACOFT) based tracking method is proposed, in our approahch the energy between the foreground and the background is adaptively balanced, which enables the target of interest always having a higher energy than its background. The samples' qualities are also evaluated in real time, which ensures that the samples used for template training are always helpful with tracking. In addition, we also propose an optimal bottom-up and top-down combined strategy for template training, which plays an important role in improving both the effectiveness and

[*] Corresponding author

Email addresses: liuqy@ciomp.ac.cn (Qiaoyuan Liu), jinchang.ren@strath.ac.uk (Jinchang Ren), wangyr915@nenu.edu.cn (Yuru Wang), 18170140658@163.com (Yuanbo Wu), sunhaijiang@126.com (Haijiang Sun), zhaohuimin@gpnu.edu.cn (Huimin Zhao)

robustness of tracking. As a result, our approach achieves a great improvement on the basis of the baseline tracker, especially under the background clutter and fast motion challenges. Extensive experiments over multiple tracking benchmarks demonstrate the superior performance of our proposed methodology in comparison to a number of the SOTA trackers.

Keywords: Visual tracking, energy-aware correlation filter(EACOFT), enhanced feature, top-down and bottom-up strategy

1. Introduction

As a hot issue in the field of computer vision, visual tracking has been widely applied into many practical applications such as intelligent driving [1], video surveillance [2], sports competition [3] et al. Great breakthroughs have been made in recent years [4], including dynamic appearance model based particle filter [5], where the correlation filter(CF)-based trackers have made an indispensable contribution [6].

The advancement of CF-based tracking performance is mainly driven by its fast calculation in the frequency domain. These trackers usually achieve accurate localizations by determining the correlation between the detetion region and the templates trained. By transforming the matching process from the spatial domain to the frequency domain, CF-trackers effectively simplify the complex matrix multiplication to a point product operation. In this way, both the tracking accuracy and the tracking speed have be improved.

The inherent limitation of CF-based trackers is that the dark targets would always show low amplitude in the frequency domain. As a result, they can hardly detect these targets especially when the background is much brighter. It is konwn that the CF-based trackers locate the target by finding the peak position on the response map. However, as shown in Figure 1, the small dark targets would inevitablely represent a rather lower energy than its background. So while calculating their response scores, the response score of the background become higher than the targets, which may seriously affect the accuracy of target

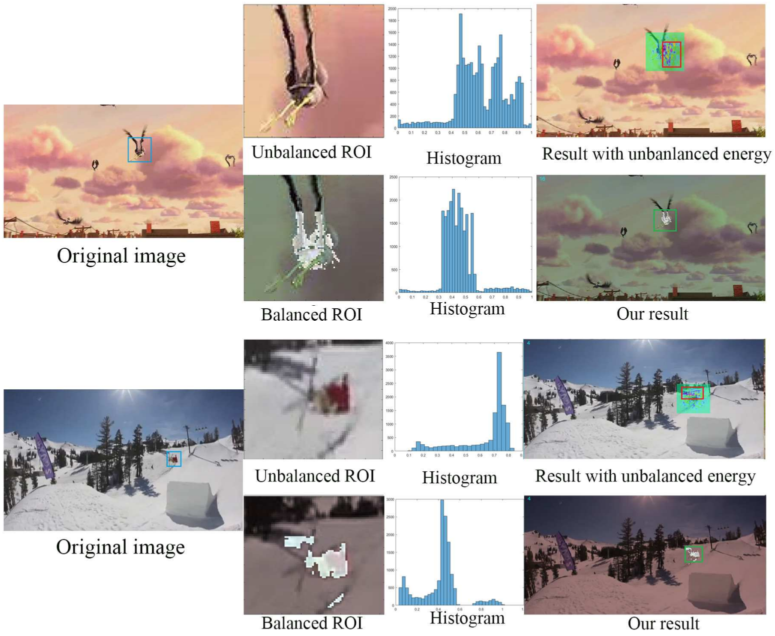


Figure 1: Image comparison before and after energy balance adjustment. 1) the baseline tracker (1st row of each group), 2) our adaptive energy balance approach (2nd row of each group)

location. This is also a common problem of all the CF-based trackers. While our approach can well overcome this drawback and achieve an accurate tracking by adaptive energy adjustment. It is also proved by the histograms in Figure 1, that after the energy balance adjustment our approach can guide the tracker to pay more attention on the target. In our opinion, the main reasons why CF-based trackers may fail to track the target in such cases can be explained as follows: i) The low-energy target would always float within a range of small values, which would make the target not significant enough in the response map difficult to be detected. ii) Mistargeting changes in the high-energy background as targets may lead to tracking drift, resulting in template deviations due to the degradation of training sample quality, and eventually tracking failures.

To address the aforementioned issues, the context-aware correlation filter tracker (CACF)[7] proposes to make the context patch less responsive than the image patch to ensure that the tracker can pay more attention to the real target. However, this method with patches was not flexible enough, leading to quite limited tracking performance. Although the idea of our energy-aware tracker is similar to the CACF tracker, our adaptive strategy based on precise contours can avoid the interference of the background for more accurate tracking. In terms of sample quality assurance, regularized correlation filter tracker with decontaminated training set (SRDCFdecon) [8] assigned an initial weight value to each sample in the training set, followed by learning the weights with the appearance model jointly, where the training set was optimized by wiping off samples with low weights. Although this method provides a good idea for the dynamic management of the sample set, it still suffers from some inaccurate tracking in various cases. Different from the SRDCFdecon tracker, in our approach, a sample set optimization strategy is proposed, where we evaluate the quality of each sample by combining the wrong sample rejection and the bottom-up sample generation strategies, so as to alleviate the impact of low-quality samples. As deep learning methods [9] pay more attention on the representation methods for tracking improvement, they are often used to obtain more effective features to improve the overall tracking accuracy. Although these methods bring some improvements, the model training process is generally very time consuming.

In this paper, a novel tracking method is proposed to overcome the drawbacks of the state-of-the-art CF based trackers. The main contributions can be summarized as follows: i) An energy-aware strategy based on precise masking is proposed for the first time to achieve a more accurate localization of the target; ii) Bottom-up and top-down strategies are combined for optimizing the CF-based tracker, which enables rejection of low quality samples and automatic recovery under complex tracking situations; iii) For better results and higher efficiency, the contributions from the top tracker ECO-HC are integrated into our scheme to further enhance the tracking performance. Comprehensive experiments on the OTB100 [10] and VOT2016 [11] datasets have fully demonstrated the superior performance of our proposed method in comparison to a number of the state-of-the-art trackers.

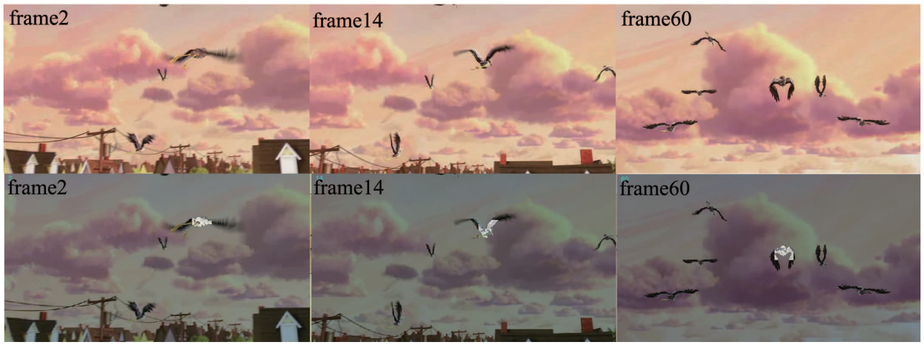


Figure 2: Comparison between original images(top row) and the actually adapted (i.e new input) images(bottom row) in consecutive frames.

2. Related work

Correlation filter based trackers: The correlation in Correlation filter(CF) is an operation to measure the similarity between two signals. If the two signals are very similar, the correlation value between them would be high, and vice versa. While in visual tracking, a template trained is used to detect the regoin of intrest and locate the maximum response as the most likely matched target.

The CF-based trackers generally have three basic steps: i) Learn a CF template based on the appearances of targets in given frames; ii) Apply the CF template on current frame to find the most responsive position to be the target, and iii) Update the correlation template according to the results obtained in Step ii). Different from the conventional binary-classification methods such as SVM (supported vector machine) [12] where the labels are either 0 or 1, the CF based trackers treat the labels of samples to be continuous to form a confidence map, and then find the position with the highest response on the map as the matched target. To separate the foreground from the background, continuous labels would be more delicate and convincing [13]. Experiments show that CF can effectively improve the accuracy and robustness of visual tracking [4],[14],[15].

CF has been used for tracking before 2010. In particular, synthetic discriminant functions [16] have been used for feature point tracking [17] and object tracking [18], [19], [20], etc. The new variations include the works of Bolme et al. [14] and Henriques et al. [21]. Although being simple and failing to perform well in all tracking difficulties, these have laid a solid foundation for further development in recent years.

Generally speaking, CF-based trackers can behave well with high efficiency. Specifically, the accuracy of the CF dominates the efficacy of tracking, while the feature calculation in Fourier domain leads to high efficiency. According to the mechanisms employed for improvement on the tracking accuracy, the existing CF-based trackers can be divided into two categories, i.e. improved tracking strategies and enriched features. These are discussed in detail as follows.

In the first category, different tracking strategies are employed to enhance the tracking performance. In Ma et al [22], Long-term correlation tracking (LCT) was proposed to handle the tracking problems with long-term occlusion. By considering temporary context information around the target, the spatial weights between the target and the background were integrated. As a result, the prior knowledge could be preserved whilst learning any new information. In [23], another representative method, spatially regularized CF for visual tracking (SRDCF), was proposed. By introducing a spatial regularization component into the general objective function, the unwanted boundary effect caused by the circular matrix in CF trackers could be reduced and result in improved tracking performance. Based on the SRDCF method, a further extension namely spatially regularized CF tracker with decontaminated training set (SRDCFdecon) was proposed [8]. After assigning an initial weight value to each sample in the training set, the weight values and the appearance model were learned jointly. In order to improve the tracking performance, the training set was optimized by discarding samples with low weights, which had led to increased tracking efficiency and decreased tracking errors.

In the second category, more effective features are emphasized to enhance the tracking performance, based on the assumption that the targets can be more easily localized when they are better described. In [24], color names (CN) was introduced into the CF trackers. After dividing the Red-Green-Blue (RGB) color space into 11 subspaces including black, blue, brown, gray, green, orange, pink, purple, red, white and yellow, the principal component analysis (PCA) algorithm was used to select the most prominent colors from each frame for real-time tracking. This adaptive method greatly reduced the dimension of features from 11 to 2, where both the tracking accuracy and tracking efficiency were improved. In Bertinetto et al [25], the Complementary Learners for Real-Time Tracking approach was proposed to combine the Histogram of Oriented Gridients (HOG) feature with the global color histogram [26]. As the HOG feature is robust to motion blur and illumination changes except deformation whilst the color histogram is sensitive to illumination but robust to the deformation caused by spatial variations, they are complementary to each other for improved tracking.

With the overwhelming of deep features [22],[27],[28],[29] being introduced into visual tracking, relevant trackers such as DeepSRDCF [28] and C-COT [29] have achieved great performance. Except for the relatively strong discriminability, there are also some drawbacks for the deep features such as high computational complexity and easily over-fitting. Considering these pros and cons, recently one of the top trackers Efficient Convolution Operators for Tracking(ECO) was proposed in [4] where both deep features and hand-crafted features including HOG and Color Names were used for tracking. Besides applying PCA for dimension reduction, to reduce the computational complexity and avoid overfitting, a compact generative sample space model was proposed to reduce the training burden via combining similar samples using the Gaussian Mixture Model (G-MM) [30]. For improved tracking speed, the frequency of model updating was reduced to a fixed number rather than updating the model in each frame. As a result, both the tracking speed and tracking accuracy were tested to be almost top either among deep trackers or hand-crafted trackers.

Deep Learning based tracker: Despite of its great success in many other object detection and recognition tasks [31], there exist great challenges in applying deep learning into visual tracking. The main problem is the difficulty in training the models, as the effectiveness of deep learning mainly comes from learning of sufficient amount of labeled data. Whilst visual tracking only provides the bounding box in the first frame as the training data, the pre-trained model can only be fine-tuned using the limited sample information in current frame, regardless the fast moving/changing of the target and the background. The tracking performance heavily depends on the quality of the pre-trained model and the quality of the training data, which undoubtedly increases the complexity and limitations of the trackers. The representative deep tracking method MDNet [27] extracts motion features for visual tracking. Although it is the winner of the championship of VOT 2015 [32] in terms of the overall accuracy, it has to buffer hundreds of proposals and results in a fairly slow tracking speed. In addition, ADNet [33] uses a variety of training sequences to pre-train the neural network, which can control the action and fine-tune the model during the process of tracking. However, it can barely achieve real-time tracking on the Graphics Processing Unit (GPU), in fact, the calculation load is too heavy to be applied in practical applications. In contrast, the correlation filtering can track targets with much less computational load, and be easier to achieve accurate and real-time tracking. It can also complete the model training only using the Central Processing Unit (CPU), not limited to the designated hardware, hence it is more suitable for practical applications.

3. The baseline tracker

Two versions are proposed in the ECO approach: deep learning version and classic hand-crafted version. Considering the tracking speed, the hand-crafted version (ECO-HC) is selected as the baseline tracker, aiming to achieve the similar performance to the deep version but with a higher efficiency. First of all, the principle of ECO is briefed as follows.

To improve the time and space efficiency of CF-based trackers, a theoretical framework for learning efficient convolution operators was proposed in ECO [4], where a factorized convolution approach was employed to reduce the size of model and the complexity of features. While tracking with high-dimensional features, updating the model every time involves about 800,000 parameters, which could not only decrease the tracking speed, but also cause over-fitting. The factorized convolution operator can be given by:

$$S_{Pf}\left\{X\right\} = (P \cdot f) * J\left\{X\right\} \tag{1}$$

where P is an coefficient matrix learned by PCA from the initial frame, which is the key factor for reducing the model size; f represents the original CF, $S_{Pf}\{X\}$ represents the response score of the training sample X on the CF template f, X and $J\{X\}$ respectively denote the training samples and the interpolated feature map as described in C-COT [29]. The optimal filters were determined by minimizing the following objective function:

$$\begin{aligned} E(f,P) &= \left\| S_{Pf}\left\{\hat{X}\right\} - \hat{y} \right\|_{l^2}^2 + \sum_{c=1}^{C} \left\| \hat{\omega} * \hat{f}^c \right\|_{l^2}^2 + \lambda \left\| P \right\|_F^2 \\ &= \left\| (J\left\{\hat{X}\right\})^T P \hat{f} - \hat{y} \right\|_{l^2}^2 + \sum_{c=1}^{C} \left\| \hat{\omega} * \hat{f}^c \right\|_{l^2}^2 + \lambda \left\| P \right\|_F^2 \end{aligned} \tag{2}$$

Under the constraints of the two regular terms for boundary effect alleviation, the difference between the correlation score map S_{Pf} and the true value y is minimized for training the CF. Variables with a hat like $\hat{y}$ denote coefficients of the Fourier series. Classic CF-based trackers are used to collect samples in continuous frames, leading to highly similar samples in the training set and templates over-fitting to adjacent frames. To solve this problem and reduce the size of the training set at the same time, a probabilistic generative model is introduced to generate a compact description in the ECO tracker, which can not only eliminate redundant samples but also enhance their varieties. Specifically, a Gaussian Mixture Model (GMM)$\sum_{l=1}^{L} = \pi_l N(X; \mu_l; I)$ is utilized to generate different training components while compacting similar samples, so as to improve the accuracy of template modeling.

$$\pi_n = \pi_k + \pi_l, \quad \mu_n = \frac{\pi_k \mu_k + \pi_l \mu_l}{\pi_k + \pi_l} \tag{3}$$

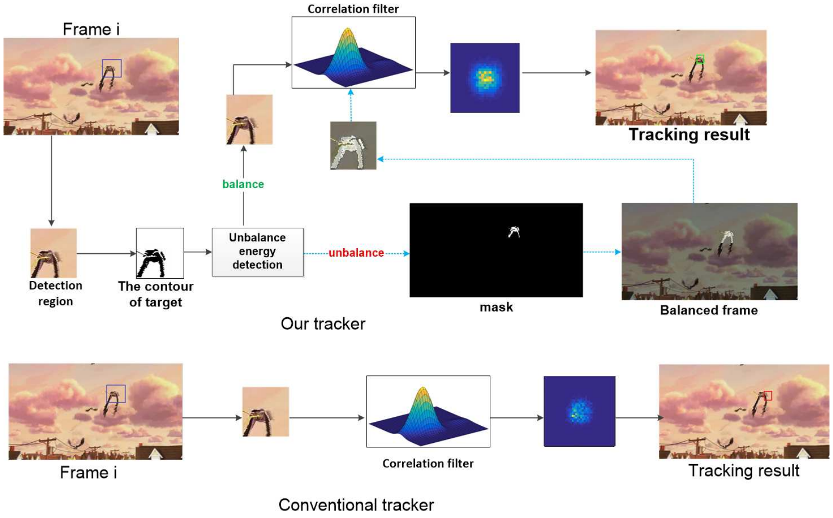


Figure 3: Comparison of our proposed energy-aware CF-based tracker (EACOFT) and the conventional tracker.

where π and μ are the prior weight and the mean value of the corresponding component l, respectively. After discarding the components whose weight are below a given threshold, two closest components k and l are merged into a new component n.

The main purpose of the process above is for increasing the tracking speed, which can be achieved by either reducing the number of CFs or the number of training samples. However, even with a high tracking speed, the aforementioned challenges in Fourier domain remains unsolved. Despite of using strong discriminant but low speed deep features, inaccurate tracking or complete loss of object would always occur while tracking low-energy targets. Based on the analysis above, two effective strategies are proposed in our approach and discussed in detail in the next section.

4. Adaptive energy-aware strategy for tracking

The energy distribution between the foreground and the background is very important for correlation filter-based tracking in the Fourier domain. The object can be easily detected when it is a bright and significant target in the dark background, i.e. a relative high-energy target. In contrast, dark targets in bright background would show a relative low-energy, as any changes may be easily flooded by the high-energy background. We call this case as the unbalanced energy distribution. Once the tracker fails to detect the changes of targets, inaccurate tracking or complete loss of targets may occur.

To tackle this problem, for the first time an energy-aware strategy based on precise masking is proposed in our approach. Once the unbalanced distribution is detected, the energy balance strategy would be adaptively applied in order to ensure that the targets can always be of a relative higher energy than the background for easy detection. The difference between the energy-aware tracking and conventional tracking is illustrated in Figure 3.

4.1. Unbalance energy detection

While tracking with the CF-based tracker, the unnoticeable targets in dark color tends to show much lower energy than its background in the frequency domain. To enable effective detection of such targets, it is especially important to make the following adaptive adjustment. When the foreground is more significant than the background, it would be more conducive to tracking. Therefore in our approach, the energy anomaly detection alerts the low-energy target in time by comparing the pixel difference between the foreground and the background. Indeed, the energy balance strategy can be implemented in other color spaces where the intensity component can be directly used. Considering the transform of the color space consumes time, we simply used the original RGB color space, where the unbalance of energy distribution can be detected as follows:

$$s = \begin{cases} 1, \mu_f(R) \leq \mu_b(R)) \wedge (\mu_f(G) \leq \mu_b(G)) \wedge (\mu_f(B) \leq \mu_b(B)) \\ \qquad 0, \text{otherwise} \end{cases} \tag{4}$$

where μ_f represents the mean value of the foreground, μ_b represents the mean value of the background. For the region detected in the previous frame, if its μ_f is lower than its μ_b, the enery unbalance state would be passed to the tracker through a label s, in this case an adaptive adjustment would be applied. While judging the difference between the foreground and the background, an adaptive threshold segmentation method [34] is used to distinguish the target from the detection region. The segmentation algorithm will generate a binary map to show the contour of the detected foreground. Accordingly, the image would be divided into two regions corresponding to the foreground and the background.

4.2. Energy balance strategy

Considering that the degree of energy unbalance varies from frame to frame, therefore, an adaptive adjustment strategy is proposed for energy balance. For the frames whose foreground is significant enough, there is no need to apply a large balance operation, while for the frames whose foreground is significantly darker than the background, it is necessary to increase the balance degree. Therefore, our approach innovatively proposed an adaptive energy balance strategy guided by the difference between the foreground and its background. First, the difference between the foreground and the background can be calculated channel-by-channel as follows

$$l(c) = \mu_b(c) - \mu_f(c), c = \{R, G, B\} \tag{5}$$

The segmentation result generated by the thresholding segmentation algorithm mentioned in the previous section can be easily converted into a binary mask,

$$M = \begin{cases} 1 & \text{(foreground)}, \mathrm{M_{i,j}} > \tau \\ 0 & \text{(background)}, \text{otherwise} \end{cases} \tag{6}$$

when the pixel $M_{i,j}$ is larger than the threshold, assign it to foreground and set its value to 1, otherwise assign it to background and set it to 0. In this way, a binary mask M could be generated (as shown in Figure 3), which will be used to balance the energy of the dark-targets. By applying the mask M to every channel of the image, the foreground can be separated precisely from the background. By adaptively brighting the foreground while suppressing the background the significance of the foreground can be effectively enhanced as follows

$$I_b{'} = (I(c) - l(c) - \delta) \times M, c = \{R, G, B\} \tag{7}$$

$$I_f{}' = (I(c) + l(c) + \delta) \times \sim M, c = \{R, G, B\} \tag{8}$$

where I denotes the original image of the current frame and δ is a constant, $\sim M$ is the negated mask. Based on the adaptively adjusted foreground and background, the background in the Fourier domain is suppressed whilst the foreground is enhanced. As a result, the frame would be updated with the following formula, aims to track the low-energy target more easily:

$$I' = I_b{}' + I_f{}' \tag{9}$$

5. Energy-aware CF based tracking

5.1. Rejection of low quality samples

Due to the complexity of visual tracking, there are always situations as shown in Figure 4 where the pixels in the bounding box of the expected target may not belong to the target or most of them are non-target, these are namely low quality samples in our paper. Since the CF-based tracker is based on collecting the previous tracking results as samples for template training, one or two low quality samples introduced into the training set in a short time may not affect the tracking performance significantly. However, accumulated drifts generated by more low quality samples will make the template tend to be inaccurate, where the tracker may probably locate to a wrong position rather than the real target. To tackle this problem, in our approach a top-down evaluation method is proposed based on the generated sample model in ECO [4], which aims to control the quality of samples in the training set.

In the proposed strategy, the quality of each sample to be integrated into the training set would be evaluated, so as to reject the low-quality samples in time. By comparing the feature similarity between the current new sample and the existing samples in the training set, a sample can be determined as a low-quality one or not. For collecting high quality samples collecting, the feature similarity is controlled by a predetermined threshold as shown in Eq. (10), in which X_{n+1}

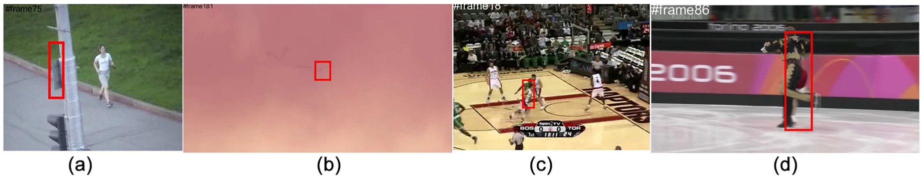


Figure 4: Different situations with low quality samples which need to be rejected: (a) The person to be tracked is completely occluded by the pole while running; (b) The bird to be tracked gets blurred by the heavy cloud; (c) The basketball player to be tracked is 80% occluded by another player; (d) The female skater to be tracked in red dress is 90% occluded by her partner.

represents the $(n+1)^{th}$ sample in the training set. If the current sample X is sufficiently similar to the samples in the training set, it will be kept, otherwise it will be rejected. The strategy for generating the new sample X' to replace the rejected one will be discussed in the next section.

$$X_{n+1} = \begin{cases} X, D_{\mathrm{m}} \leq \varepsilon_1 \\ X', \mathrm{otherwise} \end{cases} \tag{10}$$

In Eq. (10), the threshold ε_1 is the tolerance parameter to determine whether the current sample is a low quality one. Our approach adopts the L1 distance (Manhattan distance) between the current sample and all samples in the training set as its average similarity D_{m}, which can be defined as follows:

$$D_{\mathrm{m}} = \{mean(d_i) \mid d_i = ||V_X - V_{X_i}||_{l_1}, X_i \in T, i = 1, ...n\} \tag{11}$$

where V_X represents the feature vector of the new sample to be added into the training set $T = \{X_1, X_2, ...X_n\}$, and V_{X_i} denotes the feature vector of the i^{th} sample in the training set, k represents the k^{th} dimension of the feature vector, D is the total dimension of the features.

In this way, the influence due to fast motion, motion blur, out of view et al. could be solved effectively. The quality of the samples in the training set would be well guaranteed, thus the accuracy of the templates trained would also be improved accordingly.

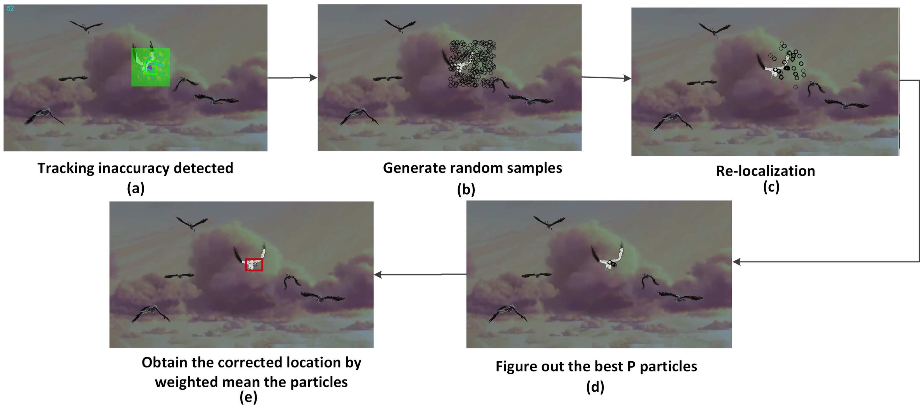

Tracking inaccuracy detected
(a)
Generate random samples
(b)
Re-localization
(c)
Obtain the corrected location by weighted mean the particles
(e)
Figure out the best P particles
(d)

Figure 5: The processing of the top-down and bottom-up optimization strategy for location correct, the black circles drawn on images represent the center locations of candidate particles

5.2. Top-down searching strategy

After rejecting the low-quality samples, a new sample needs be generated to fill the gap. In our approach, a top-down searching strategy is proposed for generating a replacement sample as detailed below.

Figure 5 illustrates the work-flow of the proposed top-down searching strategy. In most cases, inaccurate tracking is caused by the drifted tracker, hence we can assume that the real target should be somewhere nearby. Inspired by the idea of the particle filters [5], to identify the accurate location of the target, random locations considered as particles around the region of interest would be searched as shown in Figure 5(b). Since the sample information belongs to a higher-level feature than the distance information, this distance based searching strategy is namely top-down searching strategy.

We firstly evaluate and rank the mean distances between every randomly generated particle and existing samples in the training set. Particles with smaller mean distances are figured out as suitable samples for training as shown in Figure 5(d). Finally, a weighted voting scheme is employed to determine the best location of the samples as shown in Figure 5(e).

As the change of the target motion between two consecutive frames is small, we assume that the target moves at a uniform speed. The difference between the positions of the previous two targets is used as an estimate of the current motion velocity, besides a Gaussian noise δ_{rand} represents the uncertainty of target motion is added.

In total K random particles $R = \{r_j \mid r_j = X_{i-1} + p_v + \delta_{\mathrm{rand}}, j = 1, 2, ...K\}$ are generated according to the base position, the estimated velocity $p_v = X_i(x, y) - X_{i-1}(x, y)$ and the random parameter δ_{rand}, where (x, y) represents the center position of the corresponding sample. To increase the effectiveness of searching, we optimize the particle set R to S as shown in Figure 5(c) by re-localizating each particle using a correlation operation between every random position r_j and the current CF template f as shown in Eq(12), where * is the convolution operation:

$$S = \{s_j \mid s_j = r_j * f, j = 1, 2, ...K\} \tag{12}$$

For each particle s_j, its similarity to all the existing samples in the training set is compared using Eq.(13), where a similarity value $D'_{\mathrm{m}}(j)$ corresponding to the particle s_j could be obtained. For K particles in S, in total K similarity values are calculated $D_m{}' = \{D'_{\mathrm{m}}(j) \mid j = 1, 2, ...M\}$, m is within $[1, K]$, based on which, we can determine whether a usable sample can be generated from the random particles by

$$X' = \begin{cases} \emptyset, min(D_{\mathrm{m}}{}') \geq \varepsilon_2 \\ X_{\mathrm{p}}, \mathrm{otherwise} \end{cases} \tag{13}$$

Here another threshold ε_2 is set for the minimum value of $D_m{}'$ to control the suitability of the new sample, where $D_m{}'$ represents the average similiarity of the particle and the samples in the training set.

If the minimum of $D_{\mathrm{m}}{}'$ is still larger than ε_2, it means that the most similar particle to the training set is still unqualified. In other words, we can concluded that, at this particular moment, there may have no sufficient pixels in the bounding box that can be assigned to the target region, due possibly to occlusion or other impact factors. In this case, all the incorrectly tracked samples will be discarded and the X' will be set as an empty set $\emptyset$. The tracking result of the previous frame will be taken as the result for the current frame, meanwhile the CF template will not be updated until the required target reappears.

On the contrary, if the minimum of ${D_{\mathrm{m}}}'$ is less than ε_2, we will chose the samples closest to the target from the random particles to generate a new sample instead. The minimum distance from each particle to all samples in the training set can be determined as

$$E_{\mathrm{j}} = \left\{min(d_j) \mid d_j = ||V_{s_j} - V_{X_i}||_{l_1}, s_j \in S, X_i \in T, i = 1, ...n, j = 1, 2, ...K\right\} \tag{14}$$

Based on E_{j}, all the particles can be ranked. For two particles a and b, $E_{\mathrm{a}} < E_{\mathrm{b}}$ means that the particle a is closer to the target than b. In order to obtain a more reliable tracking result, as shown in Figure 5(d), in total P particles which are closest to the target are chosen to generate the final sample set.

To this end, a weighted voting scheme is introduced into the sample generation step as shown in Figure 6, where the weight of each particle is determined by

$$w_j = C - E_{\mathrm{j}}, j = 1, 2, ...M \tag{15}$$

With a constant C, a higher weight is assigned to the closer particle and a lower weight to the further particle. As a result, the location of the new sample is eventually generated as follows:

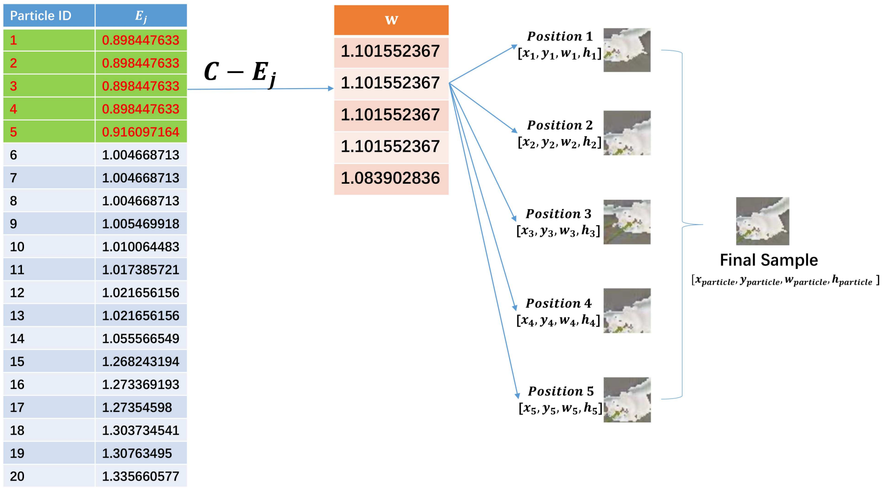
$$X' = X_{\mathrm{p}} = \frac{1}{P}\sum_{j=1}^{P} w_j \times s_j \tag{16}$$

Afterwards the new position of X_{p} is outputted as shown in Figure 5(e), which will be integrated to the training set for tracking.

5.3. Train the CF using the bottom-up strategy

For accurate tracking, a bottom-up strategy is proposed for template training. The decision for updating the upper-level filter template is made according to the quality of the bottom-level samples in the training set.

As D'_{m} represents a general quality index of the current training sample, if $min(D'_{\mathrm{m}})$ is still over the predefined threshold after re-localization, this indicates



Particle ID	E_j
1	0.898447633
2	0.898447633
3	0.898447633
4	0.898447633
5	0.916097164
6	1.004668713
7	1.004668713
8	1.004668713
9	1.005469918
10	1.010064483
11	1.017385721
12	1.021656156
13	1.021656156
14	1.055566549
15	1.268243194
16	1.273369193
17	1.27354598
18	1.303734541
19	1.30763495
20	1.335660577

w
1.101552367
1.101552367
1.101552367
1.101552367
1.083902836

$C - E_j$
$Position\ 1$ $[x_1, y_1, w_1, h_1]$
$Position\ 2$ $[x_2, y_2, w_2, h_2]$
$Position\ 3$ $[x_3, y_3, w_3, h_3]$
$Position\ 4$ $[x_4, y_4, w_4, h_4]$
$Position\ 5$ $[x_5, y_5, w_5, h_5]$
Final Sample
$[x_{particle}, y_{particle}, w_{particle}, h_{particle}]$

Figure 6: The processing of the weighted voting scheme

that all the generated particles are unsuitable. To this end, this new sample will not be added to the training set. If no new samples are added, there is no need to update the filter template, hence the reduced time in training the filter template.

Once the minimum of ${D_{\mathrm{m}}}'$ is no more than the threshold, a sample X' will be generated using the approach introduced in Section 5.2 and be integrated into the training set. We assign a weight ρ_l to every sample in the training set using Eq.(17) below, where a learning rate parameter $\eta \in (0, 1)$ is used to determine the prior weights for the L most recent frames. The weights for all frames older than frame $t - L$ are set to a constant α,

$$\rho_L = \begin{cases} \alpha, l = 1, ...t - L - 1 \\ \alpha(1-\eta)^{t-L-l}, l = t - L, ...t \end{cases} \tag{17}$$

where $\alpha = (t - L + \frac{(1-\eta)^{-L}-1}{\eta})^{-1}$ is determined by the condition $\sum_l \rho_l = 1$.

When a new sample is integrated to the training set, the previous weights ρ_l of all the samples will be updated. If the lowest weight ρ belows the threshold ε_3, the sample with the lowest weight would be replaced by X', as a result all the weights of samples would be updated using Eq.(17). Otherwise, the sample pair with the minimum distance in the training set would be extracted as $\{X_p, X_q\}$ with a distance d_1. Another sample pair $\{X', X_l\}$ is also identified, in which X_l represents the sample closest to X' in the training set, and the distance between X' and X_l is denoted as d_2.

If $d_2 < d_1$, X_l will be replaced by X', otherwise the sample with the lower weight in $\{X_p, X_q\}$ would be replaced by X', followed by weight adjustments using Eq.(17). In this way, a new template with higher quality samples is trained for the next frame, thus the tracking accuracy can also be improved whilst the training time is reduced.

5.4. Summary of the proposed tracker

In this paper, an adaptive energy-aware tracker is proposed to solve the common problem of the CF-based trackers in accurately locating the low-energy targets in the frequency domain. The energy of every coming frame is balanced adaptively for better localization after analysing the energy distribution between the foreground and the background, along with a top-down and bottom-up combined optimization strategy being introduced for more accurate correlation filtering. The overall algorithm is shown in Algorithm 1.

6. Experimental results and discussions

In this section we first introduce the experimental settings including implementation details, datasets, and evaluation metrics. We also provide both quantitative and qualitative evaluations with the baseline tracker and eleven state-of-the-art trackers as detailed below.

6.1. Experimental settings

All our experiments are tested on the MATLAB 2016 using an Intel(R) Core(TM) 2.30GHz CPU with 8GB RAM. We empirically set the parameters of our method as follows: $\delta = 30, \varepsilon_1 = 1.5, \varepsilon_2 = 2.0, C = 2.0, M = 250, P = 20$, and the other parameters are set the same as the ECO [4]. We show the effect of

Algorithm 1 Framework of the proposed tracking method

```
Input: Video frames $I_1$; $I_2$,..., $I_t$; Target state $x_0$ at the first frame.
Output: Target states $X_1$; $X_2$,..., $X_t$
  Initialization: Initialize target state $x_0$ according to the ground-truth data;

  for all time step $t$ do
    i. Calculate the energy distribution between the foreground and the background in the bounding box, and determine the $s$ as 0 or 1;
    ii. Apply the adaptive energy balance strategy if the label $s$ is 1 (Section 4.1), otherwise keep the original image;
    iii. Locate the target as $X_t$ with the correlation template T trained in the last frame, and evaluate $X_t$ with threshold $\varepsilon_1$ according to Sec. 5.1 for quality check;
    iv. Output $X_t$ if it satisfies $\varepsilon_1$ (Eq(10)), then integrate it into the training set according to Sec.5.3, next train the new correlation template $T'$ for the next frame.
    v. Generate new location ${X_t}'$ if $X_t$ doesn't satisfy $\varepsilon_1$ but satisfy $\varepsilon_2$ according to Sec. 5.2, integrate it into the training set via the top-down and bottom-up combined optimization strategy according to Sec. 5.3. Finally output ${X_t}'$ to train the new correlation template $T'$ for the next frame;
    vi. Output ${X_{t-1}}'$ and the old correlation template T for the next frame if $X_t$ is not satisfied with the aforementioned two cases.
  end for
```

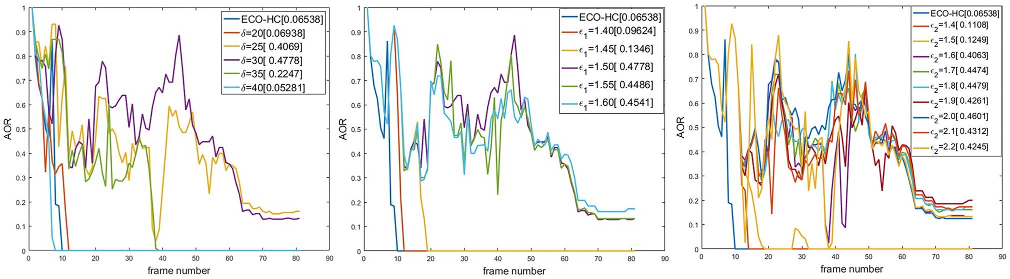


Figure 7: Overlap ratio of the video sequence “Skiing” under different parameter settings.

some key parameters including δ, ε_1 and ε_2 in Figure 7 and Figure 8. In Figure 7, different parameters are used for tracking the same video sequence “Skiing”, which is a difficult representative video from OTB100, and the optimal values of the parameters have indeed led to improved performance in the experiments. While the AOR is the average coverage between the predicted and the ground truth bounding boxes, which is a metric to measure the tracking performance frame to frame. Larger AOR represents better performance of the tracker.

As seen, δ determines the degree of the reverse energy of the target and the background. If the degree of the reversal is too small, the target couldn't be enhanced of sufficient contrast, so the energy balance strategy would be less effective. If the degree of reversal is too large, the details of the target will be lost, which will also affect the tracking performance. As shown in Figure 7, setting δ to 30 helps to yield the best results with a larger AOR and high tracking accuracy.

ε_1 measures the quality of the rejected samples. If the parameter is too small, it will lead to less representative sample set, which would be insufficient to reflect the overall change of the target. With an excessive search, the tracking speed could be affected, and it is easier to find an interference sample. On the other hand, if the parameter is too large, it may lead to more low-quality samples being integrated into the sample set, which will also reduce the discriminating ability of the model. As shown in Figure 7, setting ε_1 to 1.5 helps to gain the

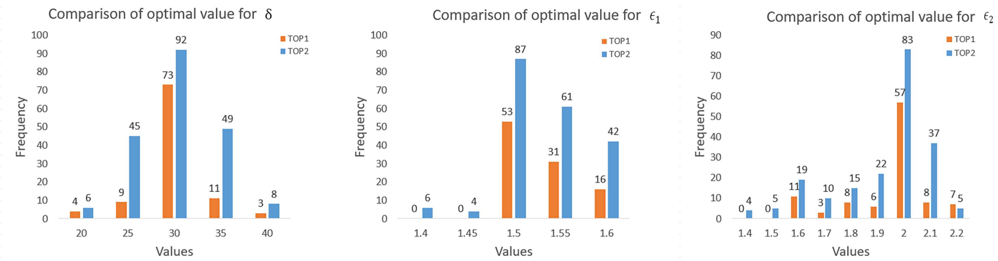


Figure 8: Comparison of the optimal parameter selection on the full dataset of OTB100.

best tracking performance.

Besides, ε_2 is the parameter to validate whether the tracker is lost. If the ε_2 is too small, the sample of the target will be filtered out. The tracker would be mistakenly considered being lost, thus it would stay at the previous location for continuous searching, and finally resulting in tracking failure. However, if the parameter is too large, the error sample will be added to the sample set, in this case the cumulative error will also make the model inaccurate and eventually lead to tracking failure. As shown in Figure 7, ε_2 is optimally set to 2.0.

Apart from this, 250 particles are generated according to the base sample. To prevent extreme jumping, the best 20 particles are chosen to generate the new sample set. We use ECO-HC as the baseline tracker due to its good performance and high efficiency. For a fair comparison, we use the same parameter settings of η, N, K and the same combination of HOG and Color Names for both our tracker and ECO-HC.

In addition to discuss the parameter selection on a single video, we also discussed the parameters on the entire OTB100 dataset, as shown in Figure 8. Here we imitate the top-k method used in deep learning for parameter selection. The probability of the selected parameters in the 100 video sequences as the optimal parameters is respectively counted. We have calculated the probability of the most optimal one and the most optimal two parameters as top-1 and top-2 for comprehensive consideration. For example, in selecting δ, there are 73 videos tested to have 30 as the top-1 value, however $\delta = 30$ is included in the results of the top-2 listed 92 videos. Finally, we determine to set 30 as the value of δ, the same tests are also applied on the selection of ε_1 and ε_2.

6.2. Datasets and Evaluation Criteria

For performance assessment, our tracking method has been extensively evaluated on the widely used OTB100 dataset [10], which includes various challenges such as illumination variation, scale variation, occlusion, deformation, in-plane rotation, out-of-plane rotation, background clutters, and low resolution. In addition, we also test our approach on the VOT2016 [11] and VOT2017 [35] datasets, which cover many representative datasets including ALOV+++ [36], non-tracking datasets, Computer Vision Online et al.

We use the success metric to evaluate all trackers tested on OTB100 datasets [10]. The success metric measures the intersection over union (IoU) of the predicted and the ground truth bounding boxes. The success plot shows the percentage of bounding boxes whose IoU score is larger than a given threshold. We use the Area under the Curve (AUC) of the success plots to rank the trackers. The precision plot is defined as the average number of frames per video that are at most 20 pixels away from the ground-truth. For a full treatment of these metrics, please refer to [10].

For the VOT16 dataset, the tracking performance is evaluated in terms of the expected average overlap (EAO), the tracking accuracy and the robustness. The EAO is based on empirically estimating the average overlap (as a function of the sequence length) and the typical-sequence-length distribution (cutting-off both lopes at a threshold such that the mass is 0.5). The measure itself is obtained as the inner product of the two functions. The accuracy is the average overlap rate of successful tracking. The robustness measures times of tracking failure in the k-th repeat. For a full treatment of these metrics, please refer to [11].

6.3. Comparison with the State-of-the-Arts

We evaluate the proposed tracker on the benchmarks mentioned above with comparison to 11 state-of-the-art trackers including ECO-HC [4], ECO [4], S-

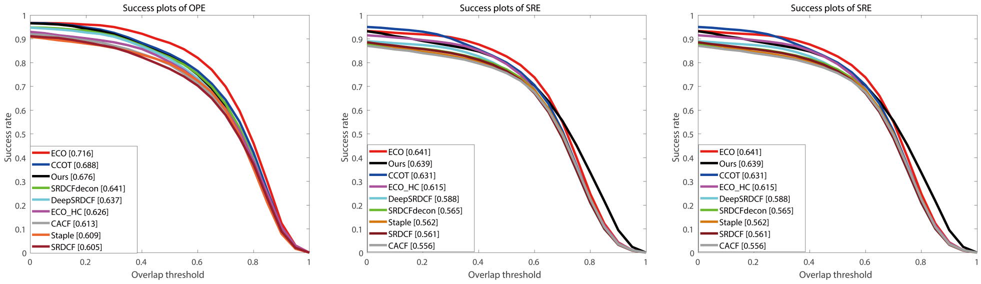


Figure 9: Average success plot of OPE, SRE and TRE on the OTB100 dataset.

RDCF [23], SRDCFdecon [8], Staple [25], DeepSRDCF [28], CCOT [29], CACF [7], UPDT [37], SACF [38], and RTINet [39], among these the approaches ECO, DeepSRDCF, CCOT, UPDT, SACF, RTINet are deep learning based trackers. In all of the three datasets, our method performs better or at least comparable to other trackers as detailed below.

6.3.1. Results on the OTB datasets

We present the success plots for OPE (one-pass evaluation), SRE (spatial robustness evaluation) and TRE (temporal robustness evaluation) on the OTB100 dataset in Figure 9. Compared to the baseline tracker ECO-HC, our tracker gains a much increasd success rate. In detail, we achieved 5.2% higher on the OTB100 than the baseline tracker in the OPE plot, 2.4% higher in the SRE plot, 1.2% higher in the TRE plot. These directly indicate the effectiveness of the strategies proposed. As the experimental results are provided in different forms, only 8 state-of-the-art trackers are compared on OTB100 dataset, our result only after the most representative method ECO and CCOT using deep learning, but we can achieve a faster tracking speed than ECO [4] which has the same framework but using deep features.

In the OTB dataset, video sequences are annotated with different attributes of tracking difficulties, which include illumination, deformation, occlusion, in-plane rotation, fast motion, out-plane rotation, scale variation and background clutter. For further comparison, we also analysed the tracking performance

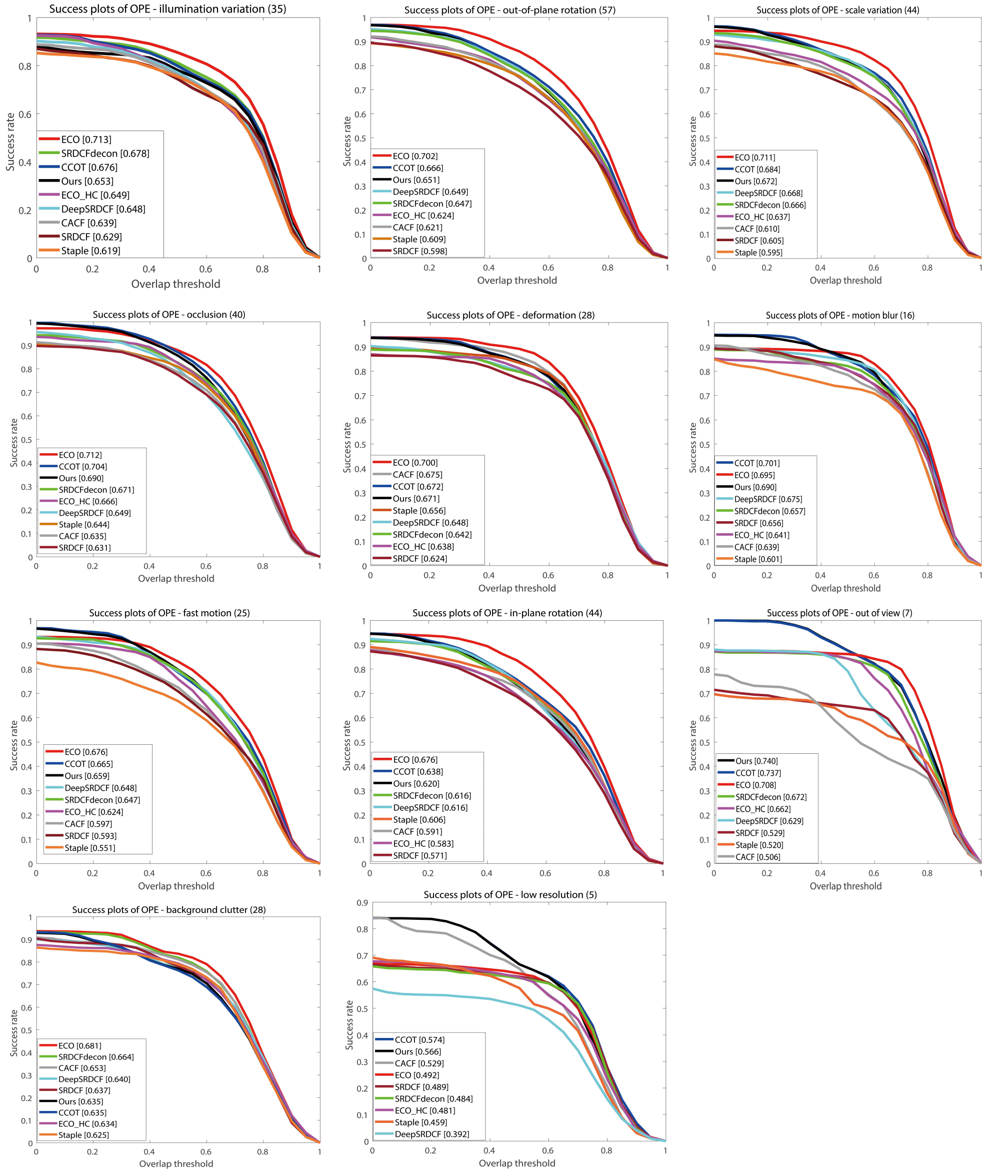


Figure 10: Attribute based evaluation using precision plots to compare our method with state-of-the-art CF based trackers on OTB100. Our method consistently outperformed almost all the trackers even the baseline method using deep features. The precision values are reported in brackets, and the number of videos for each attribute is shown in parenthesis.

under different tracking difficulties between our tracker and 8 representative trackers with the results shown in Figure 10. As seen, our tracker can well handle most tracking difficulties and achieve a great improvement compared with the ECO_HC, also a comparable performance to the deep learning based tracker such as ECO and CCOT. Thanks to the proposed adaptive energy balance and optimization strategies, our tracker can perform very well with targets made obvious and the CF learnt with updated training set in higher quality, which can successfully deal with the difficulties of occlusion, motion blur and low resolution. In this case, more accurate localization can be obtained whilst the tracking errors can be significantly reduced.

6.3.2. Comparison on the VOT dataset

Figure 11 shows the expected average overlap (EAO) on the VOT-2016 dataset for methods with publicly available implementations, among the 6 trackers compared in Figure 9. In detail the results shown in Figure 10(a) indicate a consistent trend with what we have observed from the OTB dataset, which validates the effectiveness of our method again. Our tracker is ranked the second best, almost performing the same as the 1st, ECO tracker with deep features. As seen from Figure 11(b), our approach outperforms most of the state-of-the-art trackers in different tracking scenarios especially in illumination, size change and motion change. For the non-significant improvement in occlusion, the main reason can be explained as follows: As the VOT dataset is more difficult than the OTB one, the occlusion problems are more complicated, which includes long-term occlusion, short-term occlusion, occlusion under fast-moving illumination and rotation et al. When we are rejecting the low-quality samples, a small amount of useful information may also be discarded. When ranking the performance in the category of occlusion, ECO_HC and our method respectively yielded scores of 0.204 and 0.212. Although the two figures seem quite comparable, the improvement is actually about 4%. Nevertheless, our approach has also achieved a great improvement in other categories of tracking difficulties, which have further validated the effectiveness of our proposed approach.

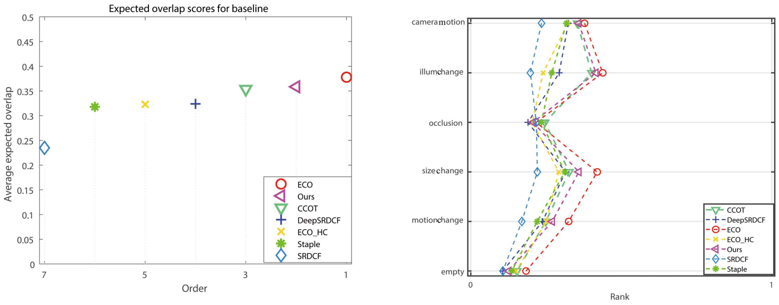


Figure 11: (a) Expected average overlap plot on VOT2016. (b) Performance rank between our approach and other state-of-the-art methods under different visual challenges.

Table 1: Analysis of our approach on the VOT2016 dataset. The impact of progressively integrating one contribution at the time is compared. We show the performance in Expected Average Overlap (EAO), Accuracy and Robustness in each step. Our contribution systematically improves both the precision and robustness.

	Baseline Sec. 3	Energy balanced strategy Sec. 4	With optimized training set Sec. 5	Our method (with both strategies)
EAO	0.322	0.331	0.346	0.350
Accuracy	0.520	0.522	0.528	0.530
Robustness	0.880	0.800	0.780	0.750

For further analysis of the performance, quantitative comparison of the results is given in Table 1. As can be seen, the integration of the energy balance strategy and training set optimization has respectively led to improved performance in terms of not only EAO but also the accuracy and robustness. As a result, when all these steps are included, the proposed tracker can perform the best with the achieved EAO, Accuracy and Robustness of 0.35, 0.53 and 0.75, respectively.

Although our approach seems to rank at the second in some tracking difficulties after the ECO with deep features, it can reach a higher tracking speed. Due to the hand-crafted features used, the dimension of the features to be calculated is much less than those using deep features. The baseline ECO-HC can operate on average at a tracking speed of 60 frames per second (FPS). However, due to the strategy of high-quality sample generation that needs to continuously filter from random particles, this has degraded the tracking speed from 60fps to 15fps. Nevertheless, this is still double faster than the deep learning based tracker ECO. As a result, our tracker can be concluded to be better or at least comparative to the top trackers. In the future work, we will further explore ways to improve the tracking speed.

In addition, we also added the comparison on the VOT2017 dataset, including the recent SOTA trackers such as UPDT, SACF and RTINet. Since these latest algorithms have no published source codes, we only compare the results given in their papers. In Table 2, we compared in detail the results of our approach and representative trackers on the VOT2016, VOT2017 and OTB100 datasets, respectively. The experimental results show that the proposed method can obtain a comparable tracking results with the recent SOTA deep learning based trackers.

Table 2: Experimental comparison between our approach and representative trackers on VOT2016, VOT2017 and OTB100 datasets.

		UPDT	SACF	RTINet	CCOT	Staple	SRDCF	ECO	ECO_HC	Ours
		[37]	[38]	[39]	[29]	[25]	[23]	[4]	[4]	
VOT2016	EAO	-	0.380	0.298	0.331	0.295	0.247	0.358	0.322	0.350
	Accuracy	-	-	0.570	0.520	0.540	0.520	0.540	0.520	0.530
	Robustness	-	-	1.070	0.850	1.350	1.500	0.720	0.880	0.750
VOT2017	EAO	0.378	-	-	0.267	-	-	0.280	-	0.273
	Accuracy	0.182	-	-	0.318	-	-	0.276	-	0.272
	Robustness	0.532	-	-	0.494	-	-	0.502	-	0.495
OTB100	Overlap precision	-	0.693	0.682	0.688	0.609	0.605	0.716	0.626	0.676

6.3.3. Qualitative comparison

For qualitative comparison, we compare in Figure 12 our tracker with the most recent five state-of-the-art trackers mentioned above on six challenging sequences with low-energy target. As seen in Figure 12(a), while after a big deformation in the sequence Bird1, the result of the tracker ECO-HC is far away from the target between frame #95 and frame #200, which can be considered as lost of target. However, our tracker can always perform well even the target is completely occluded by the cloud or re-appeared after the heavy occlusion. As shown in frame #200, most trackers suffer from an inaccurate tracking or even

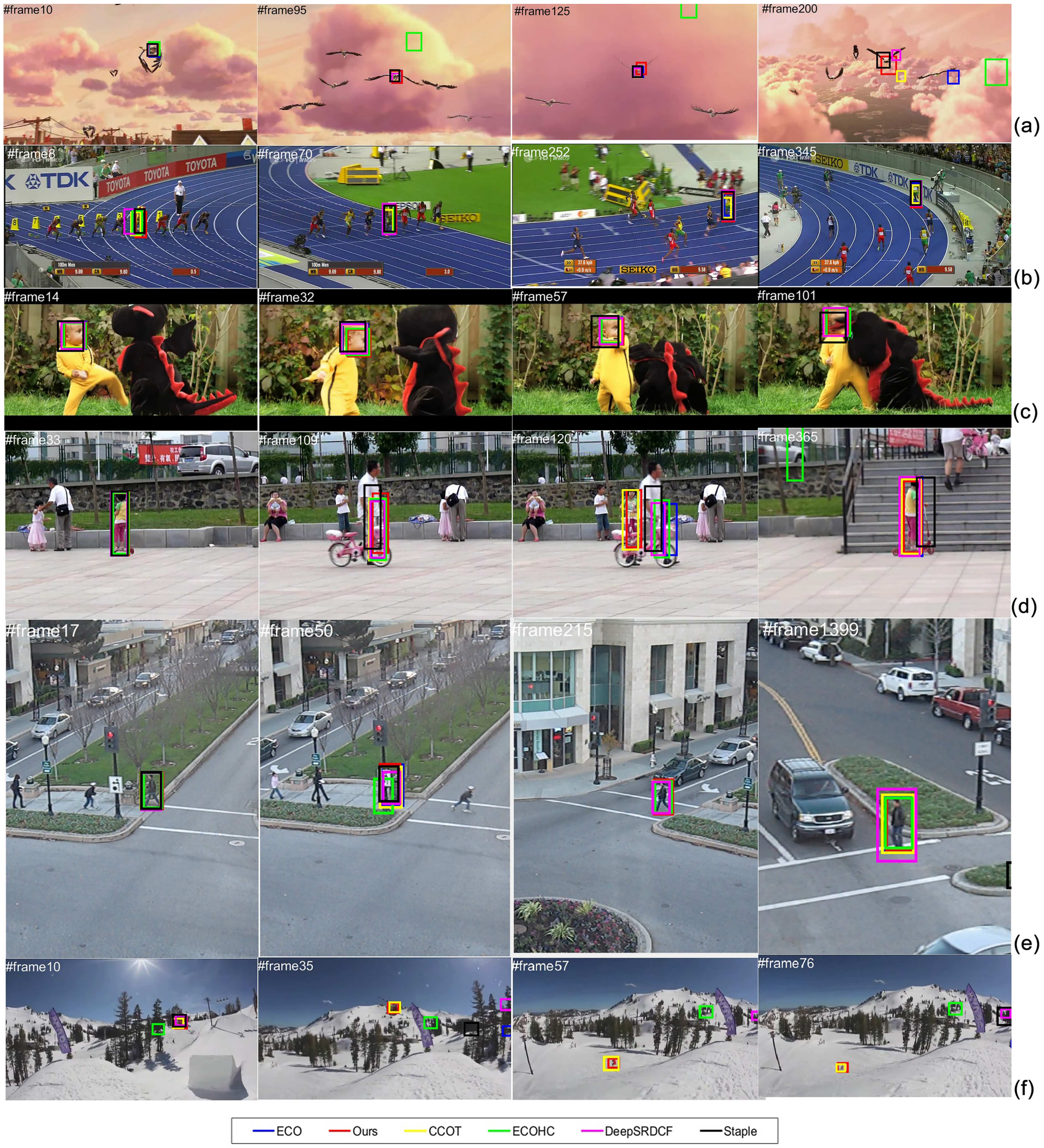


Figure 12: Comparison of the tracking results as bounding boxes in different colors for several tested videos on some key frames, where the sequence names for (a-f) are Bird1, Bolt, Dragonbaby, Girl2, Human3 and Skiing.

loss of target, yet our tracker can still accurately locate the target. Furthermore, in other occasions of occlusion such as Dragonbaby (#101), Girl2 (#120), and Human3 (#50), most trackers fail to capture the targets accurately, but our approach can still perform well. In Girl2 (#355), after a heavy occlusion only ECO, CCOT, DeepSRDCF and our tracker can capture the target. For fast motion such as Bolt (#252, #345), Dragonbaby (#32), and Skiing (#35), our tracker also shows better results than other trackers due to an accurate CF template. It is worth mentioning that in the sequence of Skiing, the target shows a relatively low energy in contrast to the background, after a severe deformation, in-plane rotation and fast motion. In frames #57 and #75, only CCOT and our tracker can still capture the target, but our results are more accurate.

The success of our algorithm can be summed up into two points: One is to deal with the unbalanced energy distribution between the foreground and the background, which can be detected automatically hence the targets can be adaptively adjusted for more accurate tracking in the Fourier domain. The second is to cope with inaccurate tracking caused by complex scenarios, and this can be detected timely by sample quality evaluation. Although low-quality samples are rejected from the training set, the tracker would continue the search until the real target is found. By doing this, the difficulty of target occlusion and re-appearance can be well handled whilst the tracking errors can also be significantly reduced.

7. Conclusion

In this paper, a novel energy-aware correlation filter (EACOFT) model is proposed for the task of visual tracking, which aims to tackle the common limitation of CF-based trackers, especially the difficulty to track low-energy targets in the Fourier domain. With the proposed energy balance strategy, the precise contour of the target of interest in the video sequence can be adaptively highlighted, enabling more effective and accurate detection, matching and tracking of the targets. By combining both the top-down and bottom-up search strategies, the proposed EACOFT can help to not only improve the accuracy of the training template but also to avoid several cases of incorrect tracking. In addition, the dynamic sample set management strategy can also help to achieve more accurate and stable tracking by correcting tracking offset in time. These two strategies have strong generalization capability in theoretically solving the inherent limitations of CF-based trackers.

As a generic solution, the proposed EACOFT can be easily applied to other CF based trackers. By combining our approach with the most representative tracker, ECO-HC as an example, comprehensive experiments have demonstrated significant improved performance on several publicly available datasets. Moreover, as the adaptive search in EACOFT will inevitably reduce the tracking speed whilst improving the tracking accuracy, how to further improve its efficiency will be focused in the future. Furthermore, the proposed model and approach can also be applied in other image matching problems, such as image classification and image retrieval, and this will also be further investigated in the near future.

8. Acknowledgement

This work is supported by the Science and Technology Development Program of Jilin Province under Grant No. 20170101144JC, the Fundamental Research Funds for the Central Universities under Grant No. 2412017FZ027 and Key Laboratory of airborne optical imaging and measurement, Chinese Academy of Sciences.

References

[1] C. E. Smith, C. A. Richards, S. A. Brandt, et al., Visual tracking for intelligent vehicle-highway systems, IEEE Trans. on Vehicular Technology 45 (1996) 744–759.

[2] J. Ren, J. Orwell, G. Jones, et al., Tracking the soccer ball using multiple fixed cameras, Computer Vision and Image Understanding 113 (2009) 633–642.

[3] J. Ren, M. Xu, J. Orwell, et al., Multi-camera video surveillance for real-time analysis and reconstruction of soccer games, Machine Vision and Applications 21 (2010) 855–863.

[4] M. Danelljan, G. Bhat, F. S. Khan, et al., Eco: Efficient convolution operators for tracking, in: In Proc. CVPR, 2017, pp. 6931–6939.

[5] Y. Wang, X. Tang, Q. Cui, Dynamic appearance model for particle filter based visual tracking, Pattern Recognition 45 (2012) 4510–4523.

[6] L. Zhang, P. N. Suganthan, Robust visual tracking via co-trained kernelized correlation filters, Pattern Recognition 69 (2017) 82–93.

[7] M. Mueller, N. Smith, B. Ghanem, Context-aware correlation filter tracking, in: In Proc. CVPR, 2017, pp. 1387–1395.

[8] M. Danelljan, G. Hger, F. S. Khan, et al., Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking, in: In Proc. CVPR, 2016, pp. 1430–1438.

[9] Y. Song, C. Ma, L. Gong, et al., Crest: Convolutional residual learning for visual tracking, in: In Proc. ICCV, 2017, pp. 2574–2583.

[10] Y. Wu, J. Lim, M. H. Yang, Object tracking benchmark, IEEE Trans. on Pattern Analysis and Machine Intelligence 37 (2015) 1834–1848.

[11] G. Roffo, M. Kristan, J. Matas, et al., The visual object tracking vot2016 challenge results, in: In Proc. ECCV, 2016, pp. 191–217.

[12] C. Saunders, M. O. Stitson, J. Weston, et al., Support vector machine, Computer Science 1 (2002) 1–28.

[13] G. Zhang, Z. Yuan, Q. Tong, et al., A novel framework for background subtraction and foreground detection, Pattern Recognition 84 (2018) 28–38.

[14] D. S. Bolme, J. R. Beveridge, B. A. Draper, et al., Visual object tracking using adaptive correlation filters, in: In Proc. CVPR, 2010, pp. 2544–2550.

[15] M. Danelljan, G. Hger, F. S. Khan, Accurate scale estimation for robust visual tracking, in: In Proc. BMVC, 2014, pp. 65.1–65.11.

[16] C. Hester, D. Casasent, Multivariant technique for multiclass pattern recognition, Applied Optics 19 (1980) 1758–1761.

[17] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, Int. J. of Computer Vision 2 (1989) 283–310.

[18] M. J. Black, Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, Int. J. of of Computer Vision 25 (1998) 23–48.

[19] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. on Pattern Analysis and Machine Intelligence 20 (1998) 1025–1039.

[20] F. Jurie, M. Dhome, Hyperplane approximation for template matching, IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (2002) 996–1000.

[21] J. F. Henriques, R. Caseiro, P. Martins, et al., High-speed tracking with kernelized correlation filters, IEEE Trans. on Pattern Analysis and Machine Intelligence 37 (2015) 583–596.

[22] C. Ma, X. Yang, C. Zhang, et al., Long-term correlation tracking, in: In Proc. CVPR, 2015, pp. 5388–5396.

[23] M. Danelljan, G. Hger, F. S. Khan, et al., Learning spatially regularized correlation filters for visual tracking, in: In Proc. ICCV, 2015, pp. 4310–4318.

[24] M. Danelljan, F. S. Khan, M. Felsberg, et al., Adaptive color attributes for real-time visual tracking, in: In Proc. CVPR, 2014, pp. 1090–1097.

[25] L. Bertinetto, J. Valmadre, S. Golodetz, et al., Staple: Complementary learners for real-time tracking, in: In Proc. CVPR, 2016, pp. 1401–1409.

[26] H. Zeng, N. Peng, Z. Yu, et al., Visual tracking using multi-channel correlation filters, in: In Proc. Digital Signal Processing, 2015, pp. 211–214.

[27] Z. Zhang, Y. Xie, F. Xing, et al., Mdnet: A semantically and visually interpretable medical image diagnosis network, in: In Proc. CVPR, 2017, pp. 3549–3557.

[28] M. Danelljan, G. Hger, F. S. Khan, et al., Convolutional features for correlation filter based visual tracking, in: In Proc. ICCV, 2015, pp. 621–629.

[29] M. Danelljan, A. Robinson, F. S. Khan, et al., Beyond correlation filters: Learning continuous convolution operators for visual tracking, in: In Proc. ECCV, 2016, pp. 472–488.

[30] Z. K. Huang, K. W. Chau., A new image thresholding method based on gaussian mixture model, Applied Mathematics and Computation 205 (2008) 899–907.

[31] W. Ma, Y. Wu, F. Cen, et al., Mdfn: Multi-scale deep feature learning network for object detection, Pattern Recognition 100 (2019) 107–149.

[32] M. Kristan, J. Matas, A. Leonardis, et al., The visual object tracking vot2015 challenge results, in: In Proc. ICCV, 2015, pp. 1–23.

[33] S. Yun, J. Choi, Y. Yoo, et al., Action-decision networks for visual tracking with deep reinforcement learning, in: In Proc. CVPR, 2017, pp. 2711–2720.

[34] X. Xu, S. Xu, L. Jin, et al., Characteristic analysis of otsu threshold and its applications, Pattern Recognition Letters 32 (2011) 956–961.

[35] M. Kristan, A. Leonardis, J. Matas, et al., The visual object tracking vot2017 challenge results, in: In Proc. ICCV, 2017, pp. 1949–1972.

[36] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, et al., Visual tracking: An experimental survey, IEEE Trans. on Pattern Analysis and Machine Intelligence 36 (2014) 1442–1468.

[37] G. Bhat, J. Johnander, M. Danelljan, et al., Unveiling the power of deep tracking, In Proc. ECCV (2018) 1804.06833.

[38] M. Zhang, Q. Wang, J. Xing, et al., Visual tracking via spatially aligned correlation filters network, in: In Proc. ECCV, 2018, pp. 484–500.

[39] Y. Yao, X. Wu, L. Zhang, et al., Joint representation and truncated inference learning for correlation filter based tracking., In Proc. ECCV (2018) 560–575.