Journal of Systems Engineering and Electronics
Vol. 35, No. 3, June 2024, pp.741 – 752

Real-time tracking of fast-moving object in occlusion scene

LI Yuran[1,2], LI Yichen[1,2], ZHANG Monan[1,2], YU Wenbin[1,2,*], and GUAN Xinping[1,2]

1. Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China;
2. Key Laboratory of Systems Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

Abstract: Tracking the fast-moving object in occlusion situations is an important research topic in computer vision. Despite numerous notable contributions have been made in this field, few of them simultaneously incorporate both object's extrinsic features and intrinsic motion patterns into their methodologies, thereby restricting the potential for tracking accuracy improvement. In this paper, on the basis of efficient convolution operators (ECO) model, a speed-accuracy-balanced model is put forward. This model uses the simple correlation filter to track the object in real-time, and adopts the sophisticated deep-learning neural network to extract high-level features to train a more complex filter correcting the tracking mistakes, when the tracking state is judged to be poor. Furthermore, in the context of scenarios involving regular fast-moving, a motion model based on Kalman filter is designed which greatly promotes the tracking stability, because this motion model could predict the object's future location from its previous movement pattern. Additionally, instead of periodically updating our tracking model and training samples, a constrained condition for updating is proposed, which effectively mitigates contamination to the tracker from the background and undesirable samples avoiding model degradation when occlusion happens. From comprehensive experiments, our tracking model obtains better performance than ECO on object tracking benchmark 2015 (OTB100), and improves the area under curve (AUC) by about 8% and 32% compared with ECO, in the scenarios of fast-moving and occlusion on our own collected dataset.

Keywords: speed-accuracy balanced, motion modeling, constrained updater.

DOI: 10.23919/JSEE.2024.000058

1. Introduction

Object tracking is a hot topic in current computer vision research field. A lot of tracking methods have been put forward since the 1970s. They were widely applied in various categories, like target tracking, surveillance, and localization. This research topic still has plenty of difficulties waiting to be solved, like the demand of real-time, high precision, stability, and resistance to interference.

Early typical models based on probability theory are particle filter, mean shift [1] and Kalman filter [2]. Kalman filter builds a motion model for the system, then adopts the main color of the object in hue-saturation-intensity (HSI) color space as its detection feature and takes it as observation input to achieve robust tracking in real world. These kinds of models are mature and complete, but in complex circumstances, their tracking result is not stable and as time goes by degradation would occur in these trackers [3,4]. As for the correlation filter [5,6], the most famous model is kernelized correlation filters (KCF) [7] which on the basis of minimum output sum of squared error (MOSSE) filter [8] introduces cyclic matrix, kernel ridge regression thus could process almost 90 frames per second and has high accuracy. Based on KCF, later scientists proposed sum of template and pixel-wise learners (staple) [9], long-term correlation tracking (LCT) [10], spatially regularized discriminative correlation filters (SRDCF) [11] and efficient convolution operators (ECO) [12]. LCT decomposes the task of tracking into translation and scale estimation, and trains a fern classifier to relocate the missing object. SRDCF further refrains the boundary effect by expanding the padding, but the time cost for this approach is too high and not suitable for real-time application. Among these methods, ECO demonstrates superiority in terms of both speed and accuracy, attributed to its implementation of mixtrue of Gaussians (MOG) model for training samples compressing and principal component analysis (PCA) for features dimension reduction effectively preserving critical information [13,14].

The rapid advancement of deep learning has led to the widespread adoption of convolutional neural networks (CNNs) across diverse research areas, including object tracking [15]. The first outstanding tracking model applied CNN is fully-convolutional siamese network (SiamFC) [16] which is trained end-to-end on the dataset and conducts stochastic gradient descent online to adapt the weights, owing high tracking accuracy in short-term.

Manuscript received October 27, 2023.
*Corresponding author.
This work was supported by the National Nature Science Foundation of China (62373246;62203299), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (SL2022MS008;SL2020ZD206; SL2022MS010).

Then, some models based on region-based convolutional neural network (RCNN) is proposed, such as Siam RCNN [17], SiamRPN++ [18]. Later, inspired by the natural language processing (NLP) model, scientists started to adopt transformers [19] to analyze video frame sequences and track objects. Some of those representative models are like accurate tracking by overlap maximization (ATOM) [20] and learning discriminative model prediction (DiMP-50) [21]. these models now gain the best performance on most mainstream public datasets, such as object tracking benchmark (OTB) [22,23], visual object tracking (VOT) [24], and unmanned aerial vehicle 123 (UAV123) [25]. Though having so many advantages, these models have much higher hardware requirements for running in real-time, and need extensive training datasets which makes them poor in portability and daily application. There are hardly any papers that discuss the combination of correlation filters and CNNs to complement each other.

Those approaches mentioned above mainly focus on feature extraction and observation. Nevertheless, when tracking, the search area is also a crucial factor to determine both the tracking speed and accuracy, especially in fast-moving scenarios. A small and precise search area could not only greatly reduce the computation cost, but also boost the tracking accuracy for blocking out the background and irrelevant interference. Additionally when missing the target, for example in occlusion scene, a reasonable search area strategy would help relocate. In [26], Jiang put forward a fixed-scale sliding window search in short-term missing and a probability multi-scale search in long-term missing strategy to relocate the lost object. While in [27], Ma et al. claimed that the performance of the tracker is sensitive to the padding size of the target's neighboring context, such as the scaling factor and aspect ratio. Yang et al. [28] came up with an idea of optimizing the search strategy by respectively calculating the mean value and standard deviation of the surrounding image patch and discarding most patches with adaptive threshold to tremendously reduce the computation cost. However, none of these papers discussed how to find the best search area in general cases, and this is a crucial issue for object tracking.

Additionally, most tracking methods update their trackers at every frame like learning continuous convolution operators for visual tracking (C-COT) [29], or at fixed N frames periodically like ECO [12], regardless of the tracking state. In this way, when the trackers are in very bad situations, the tracker would be polluted by those poor samples and their function will be degraded. For example, when the tracked object is missing or occluded, if the tracker still updates, the background or occluded stuff would be mistakenly taken as the sample of tracked object. Li et al. [30] designed a check mechanism to analyze the correlate response distribution and decide whether to renew the samples or not. Later, Jiang et al. [26] gave a more complicated confidence score to judge the reliability of tracking output. Though multiple criteria to determine the tracking state are given, the evaluation standard still could be advanced to become more efficient.

In our paper, a lightweight tracking model integrating the advantages of those approaches introduced above is proposed. Firstly, a speed-accuracy-balanced feature extraction model is put forward which could take advantage of both the high computation speed of correlation filter and high accuracy of high-level features obtained by CNNs. Then, the proposed tracking model adds Kalman filter based motion model to predict the fast-moving object's location in the future, therefore, providing a more precise search area for the tracker. Additionally, a constrained updater is designed which ensures that the tracker and training samples would not be contaminated in those terrible situations. Compared with advanced tracking models, our tracking model achieves a favorable compromise between accuracy and speed, and effectively solves the fast-moving and occlusion challenges.

Our main contributions are as follows:

(i) A speed-accuracy-balanced model is proposed. This model not requiring high-level hardware equipment can run on Intel CPU at 21 FPS and outperforms traditional correlation filter trackers in terms of accuracy.

(ii) A kinematic model based on Kalman filter is specially designed. This model provides a precise predicted search area for the tracker, making the computation speed accelerated and the tracking accuracy improved.

(iii) An updating constraint is added. This constraint prevents model degradation in occlusion scene.

The rest of the paper is organized as follows. In Section 2, preliminaries are introduced. Then in Section 3, four models: observation state judgment model, speed-accuracy-balanced feature extraction model, Kalman filter based motion model and constrained model updater that help improve the tracking speed and accuracy are put forward respectively. Experiments demonstrating the effectiveness of our model are in Section 4. Finally, we conclude in Section 5.

2. Preliminaries

Among numerous correlation filter models, ECO is the one possessing the best performance. And the proposed tracking model is set up by adding motion model, speed-accuracy-balanced feature extraction model and constrained updater on ECO.

ECO is mainly composed by two parts: correlation filter and effective convolution. The input of correlation filter is feature extracted from image which first initializes the filter, then the filter would search the area that best

matches the input samples taking it as the tracking result, and insert new samples gained from current image to update the filter. The computation procedure is as follows:

$$g = f \otimes h \tag{1}$$

where $\otimes$ means Fourier convolution. By doing Fourier transform the expression can be derived as

$$\mathcal{F}(g) = \mathcal{F}(f \otimes h) = \mathcal{F}(f) \odot \mathcal{F}(h)^*, \tag{2}$$

$$G = F \odot H^*, \tag{3}$$

where $\odot$ means dot product.

For every sample image f_i and filter h, there is corresoponding g_i, transform (3) into (4) as the correlation filter update formula.

$$H_i^* = \frac{G_i}{F_i} \tag{4}$$

To minimize the deviation between samples and object: $H^* \odot F_i - G_i$, we calculate the expression:

$$\frac{\partial}{\partial H^*} \sum_{i=1}^{M} |H^* \odot F_i - G_i|^2 = 0 \tag{5}$$

to derive eventual update formula for filter as

$$H^* = \frac{\sum_{i=1}^{M} F_i \odot G_i^*}{\sum_{i=1}^{M} F_i \odot F_i^*}. \tag{6}$$

Finally by doing inverse Fourier transform, we gain filter accomplish updating.

As for the effective convolution, ECO on the basis of C-COT introduces a factorized convolution approach which greatly reduces computation complexity. For input image f_i, ECO extracts D kinds of feature as $f_i^1, f_i^2, \cdots, f_i^D$. The feature map corresponding to these D kinds of features has N types of resolution. Using $d \in (0, D]$ and $n \in (0, N]$ to notate one of the features and resolution, the feature function could be written as $f_i^d[n]$, and for every feature d, operator J_d could be taken to integrate feature map of different resolution into continuous spatial domain, as follows:

$$J_d\{f^d\}(t) = \sum_{n=1}^{N} f^d[n] b_d \left(t - \frac{T}{N} n\right). \tag{7}$$

Thus, the correlation response of filter h on feature d is $g^d = h^d \cdot J_d\{f^d\}(t)$. For all D features extracted from sample f_i, their continuous respond function is

$$G_f\{f\} = \sum_{d=1}^{D} h^d \cdot J_d\{f^d\}. \tag{8}$$

ECO considers that all D features actually have different contributions to object tracking. Assuming C ($C < D$) features in sample play the decisive role, a matrix $\boldsymbol{P}_{D \times C}$ could reduce the dimension of feature map by $J_d\{f\} = \boldsymbol{P}^{\mathrm{T}} J_d\{f^d\}$. As a result, the features are greatly integrated and effectively used, accomplishing high processing speed.

3. Real-time tracking method for fast-moving object in occlusion scene

According to [31], a tracking method consists of the following five parts: observation model, feature extractor, model updater, motion model, and ensemble post-processor. Under this guideline, our tracking method, making use of the correlation filter, effective convolution, and feature dimension reduction strategy in ECO, sets up our model in four parts: observation state judgment model, speed-accuracy-balanced feature extraction model, motion model, and updater. The overall process of the proposed tracking method is shown in Fig. 1.

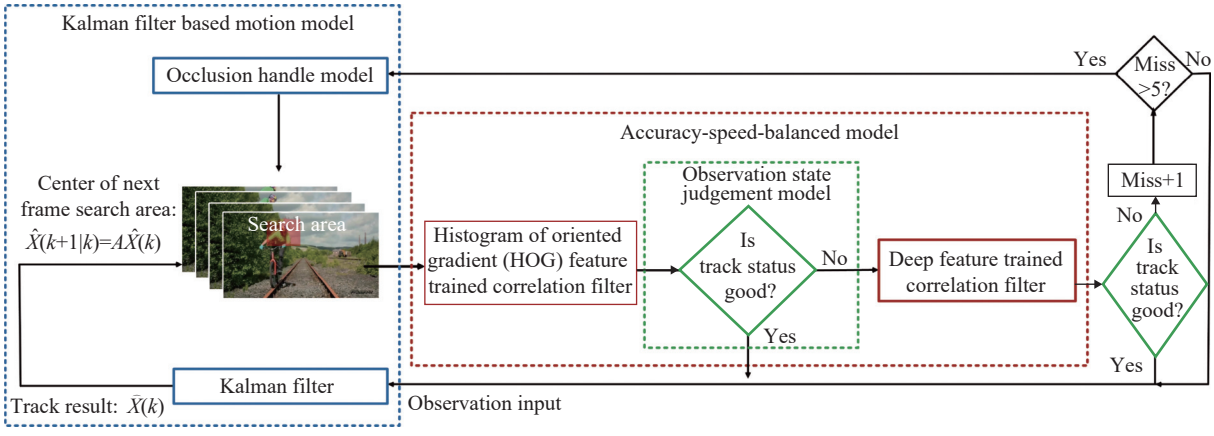


Fig. 1 Overall process of our tracking method

As shown in Fig. 1, our proposed tracking method is composed by track status judgment model, accuracy-speed-balanced model, and Kalman filter based occlusion handle motion model. The accuracy-speed-balanced feature extraction model in the red dashed box is composed of a HOG feature trained correlation filter and a deep feature trained correlation filter, additionally with an observation state judgment model within the green dashed box. The Kalman filter based motion model is in the blue dashed box. In the following sections, every part of the tracking models will be discussed in detail.

3.1 Observation state judgment model

When tracking, the position with the highest response score is regarded as the position of the target. Response score is the output of the correlation filter h in tracking model, with feature map f as input. Response score can be calculated as $f \otimes h$. However, in some cases, the tracker's performance is poor and the tracking result is untrusted, if we do not make a judgment on this situation, the probability of missing or tracking the wrong target would greatly increase. Therefore in this section, a simple and effective observation state judgment model would be discussed.

From Fig. 2(a), we can see that when the tracking result is trustworthy, the peak value is salient and much greater than its neighboring value and with little noise, while in Fig. 2(b), there is no obvious peak value, and a lot of noise exists in the response.

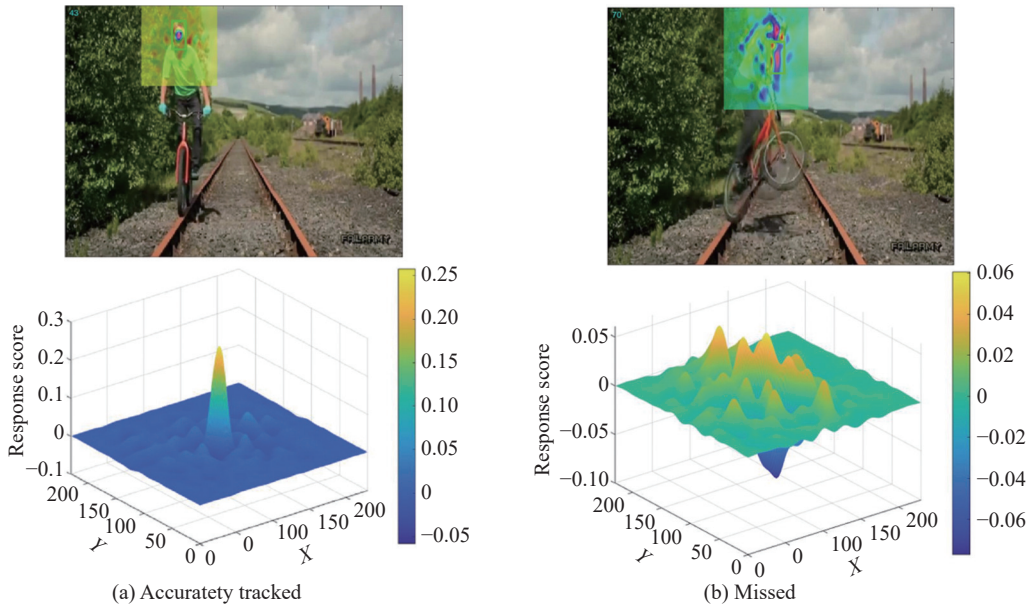


Fig. 2 Response scores in different tracking states

Based on the characteristic of response scores in different tracking states, we assume that when the medium value of the response score is greater than the mean value, the tracking result is reliable. And to make this criterion more general, we represent it as follows:

$$\text{TrackState} = \begin{cases} 1, & \text{medium} \cdot T > \text{mean} \\ 0, & \text{medium} \cdot T \leqslant \text{mean} \end{cases} \tag{9}$$

where $\text{medium} = (\max(\text{scores}) + \min(\text{scores}))/2$, mean is the average score, $\text{TrackState} = 1$ means the tracking result is unreliable, $\text{TrackState} = 0$ means reliable, T is a hyper-parameter which could manually adjust from 1 to 0.

3.2 Speed-accuracy-balanced feature extraction model

Current trackers have difficulty in finding the balance between speed and accuracy, those who run fast always adopt simpler feature extraction strategy or model construction, meanwhile, those with high accuracy always have heavier model structures and run much slower. In this subsection, a speed-accuracy-balanced feature extraction model is put forward.

On the basis of Subsection 3.1, we now have an observation state judgment model, thus when the observation state judgment model determines that the tracking state is good, we could just use a simple feature extraction approach to track and train a simple and fast tracker. When the observation state judgment model finds the simple tracker's tracking result is unreliable, the model will be changed into using a complex feature extraction method and train a more complex tracker with high accuracy. In this way, unnecessary complex computation is avoided and when the tracking state gets bad, it could be timely corrected. Fig. 3 shows the speed-accuracy-balanced feature extraction model. When the tracking state is good, HOG feature trained correlation filter is adopted. When tracking state is judged to be poor, deep feature trained correlation filter will be adopted.

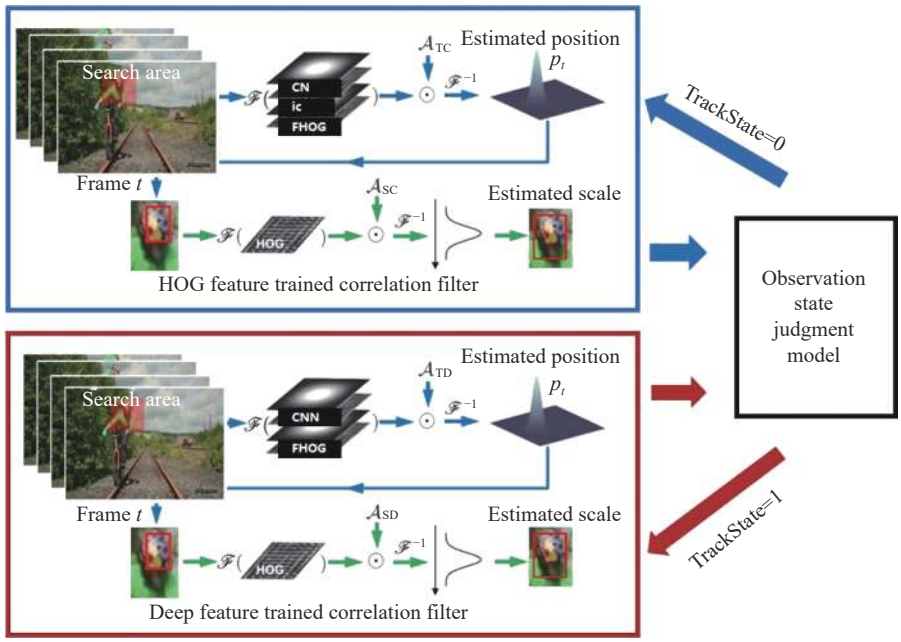


Fig. 3 Speed-accuracy-balanced model

We parallelly train two trackers, the one using histograms of oriented gradients with floating-point precision (FHOG), color names (CN) and indensity channel (ic) features in samples to train a simpler correlation filter in a lower dimension, the other using FHOG and convolution neural network extracted features to train a more sophisticated tracker in a higher dimension. When the observation state from the simple tracker is poor, the sophisticated tracker will be applied to track which requires longer calculation time to timely correct the mistake.

3.3 Kalman filter based motion model

For a fast-moving object, its position would be far away from its location in the last frame. If we assume the search area as a rectangle centered at last frame object position, the search area has to be large enough to include the object in next frame. However, by doing so, the computation cost will greatly increase and more disruptors will be included which are negative to our tracking.

In this subsection, a Kalman filter based motion model is proposed using the motion information of the object in the last frame to predict its future location in the next frame, therefore giving a more reasonable search area to tracker to obtain more excellent performance. Besides, this motion model could handle the occlusion cases, by supposing the object will follow its previous motion mode in those occluded frames. This assumption works well for object with regular movement.

Process equation: because the camera sampling rate (frame rate) is extremely high, using the idea of differential approximation, the movement pattern between frames could be deemed as uniform linear motion. On top of this, the transition matrix of the Kalman filter can be written as

$$\boldsymbol{X}(k)=\boldsymbol{A}\boldsymbol{X}(k-1)+\boldsymbol{\omega}(k-1)$$

$$\begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix}=\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x(k-1) \\ y(k-1) \\ v_x(k-1) \\ v_y(k-1) \end{bmatrix}+\boldsymbol{\omega}(k-1) \tag{10}$$

where $x(k), y(k), v_x(k), v_y(k)$ mean the pixel horizontal ordinate, vertical ordinate, horizontal speed, vertical speed at frame k respectively. Vector $\boldsymbol{\omega}(k-1)$ is the Gaussian noise with normal probability distribution $p(\omega)\sim \mathrm{N}(0,\boldsymbol{Q})$. Due to the reason that external input is changeful in various circumstances and it is impossible to find a unified expression, the external input is ignored. The state prediction equation is also as shown in (10).

Measurement equation: State space of $\boldsymbol{Z}(k)$ includes pixel horizontal ordinate and vertical ordinate, thus the measurement equation is as follows:

$$\boldsymbol{Z}(k)=\boldsymbol{H}\boldsymbol{X}(k)+\boldsymbol{v}(k)=\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix}+\boldsymbol{v}(k) \tag{11}$$

where $\boldsymbol{H}$ is the measurement matrix, $\boldsymbol{Z}(k)$ is the measurement observed at time k, $\boldsymbol{X}(k)$ is the object's state at time k, and $\boldsymbol{v}(k)$ is the Gaussian measurement noise with normal probability distribution $p(v)\sim \mathrm{N}(0,R)$.

Time update equations: The time update equations are shown as follows:

$$\hat{\boldsymbol{X}}(k|k-1)=\boldsymbol{A}\hat{\boldsymbol{X}}(k-1)+\boldsymbol{\omega}(k), \tag{12}$$

$$\boldsymbol{P}(k|k-1)=\boldsymbol{A}\boldsymbol{P}(k-1)\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{Q}, \tag{13}$$

where $\boldsymbol{P}(k-1)$ is the covariance error matrix at time k−1, and $\boldsymbol{P}(k|k-1)$ is the time update of $\boldsymbol{P}(k-1)$.

Measurement update equations: The final measurement equation of Kalman filter is as follows:

$$\boldsymbol{K}(k)=\boldsymbol{P}(k|k-1)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}(k|k-1)\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}, \tag{14}$$

$$\hat{\boldsymbol{X}}(k)=\hat{\boldsymbol{X}}(k|k-1)+\boldsymbol{K}(k)[\boldsymbol{Z}(k)-\boldsymbol{H}\hat{\boldsymbol{X}}(k|k-1)], \tag{15}$$

$$\boldsymbol{P}(k)=(1-\boldsymbol{K}(k)\boldsymbol{H})\boldsymbol{P}(k|k-1), \tag{16}$$

where $\boldsymbol{K}(k)$ is Kalman gain which could be attained by independently calculating the covariance matrix, process noise, and observation noise. $\boldsymbol{X}(k)$ is the final estimated state at time k by the Kalman filter. The time and measurement equations are calculated recursively with previous estimates to predict new estimates.

The overall process of this motion model can be seen in Fig. 4. Moreover, the pseudo is shown in Algorithm 1. When the object is judged to be occluded, the object's movement state keeps the same as the previous frame. The occlusion judgment criteria have been written in the overall process. When occlusion happens, the state transition computation result will be directly taken as the tracking result, and based on this, using the state transition equation derives the next frame search area.

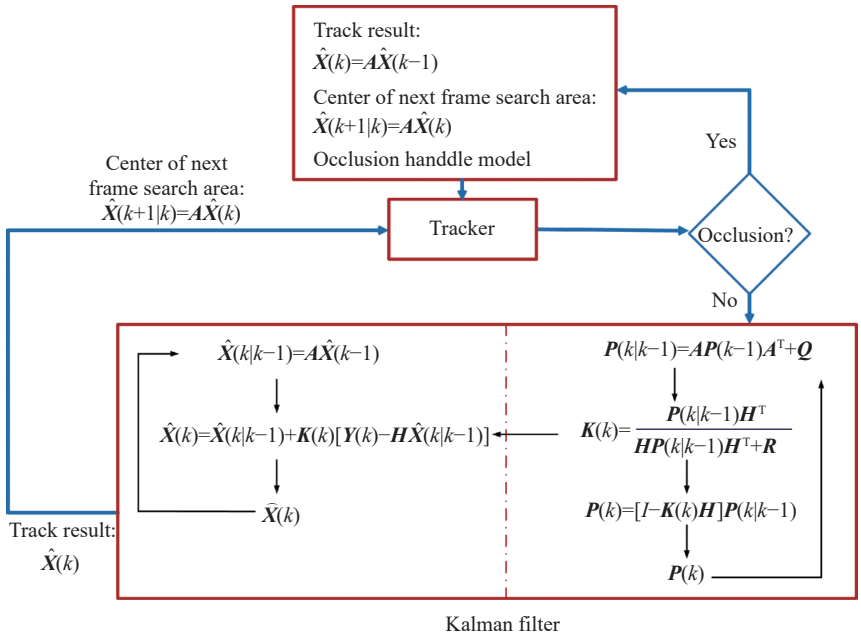


Fig. 4 Overall process of the Kalman filter based occlusion handle motion model

Algorithm 1 Kalman filter based motion model

Input: $\hat{\boldsymbol{X}}(k-1)$, $\boldsymbol{P}(k-1)$, $\boldsymbol{Y}(k)$, $\boldsymbol{Q}$, $\boldsymbol{R}$, Miss

Output: $\hat{\boldsymbol{X}}(k)$, $\boldsymbol{P}(k)$

1: Initialize $\boldsymbol{A}$ matrix and $\boldsymbol{H}$ matrix

2: **if** Miss>5 **then**

3: **goto** occlusion handdle.

4: **end if**

5: Predictstate vector and covariance:

6: $\hat{\boldsymbol{X}}(k|k-1)=\boldsymbol{A}\hat{\boldsymbol{X}}(k-1)$

7: $\boldsymbol{P}(k|k-1)=\boldsymbol{A}\boldsymbol{P}(k-1)\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{Q}$

8: Compute Kalman gain factor:

9: $\boldsymbol{K}(k)=\boldsymbol{P}(k|k-1)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}(k|k-1)\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1}$

10: Correction based on observation:

11: $\hat{\boldsymbol{X}}(k)=\hat{\boldsymbol{X}}(k|k-1)+\boldsymbol{K}(k)[Z(k)-\boldsymbol{H}\hat{\boldsymbol{X}}(k|k-1)]$

12: $\boldsymbol{P}(k)=(1-\boldsymbol{K}(k)\boldsymbol{H})\boldsymbol{P}(k|k-1)$

13: **return** $\hat{\boldsymbol{X}}(k)$, $\boldsymbol{P}(k)$

14: occlusion handdle:

15: $\hat{\boldsymbol{X}}(k)=\boldsymbol{A}\hat{\boldsymbol{X}}(k-1)$

16: $\boldsymbol{P}(k)=\boldsymbol{A}\boldsymbol{P}(k-1)\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{Q}$

17: Miss = 0

18: **return** $\hat{\boldsymbol{X}}(k)$, $\boldsymbol{P}(k)$

3.4 Constrained model updater

In this subsection, a constrained model updater is put forward aiming to solve the problem of sample and filter contamination in bad situations, like occlusion or missing. The tracker only updates its parameter and renews its

samples when the observation state judgment model claims the tracking state is reliable. This update strategy is as follows:

$$H^*(k)=\begin{cases}H^*(k-1), & \text{TrackState}=1\\ \dfrac{\sum_{i}^{M}F_i\odot G_i^*}{\sum_{i=1}^{M}F_i\odot F_i^*}, & \text{TrackState}=0\end{cases} \tag{17}$$

where $H(k)$ represents the correlation filter at frame k. When TrackState is bad, the correlation filter does not update and takes the same value as the filter of previous frame. Only when the TrackState is good, the new sample of the current frame is used to update the filter.

By taking this update strategy, when missing or occlusion happens, the original correct sample and filter are maintained, and could better relocate the object when it reappears. Otherwise, for example, when a tracking object is occluded by a board, the board would be taken as new samples and the filter would learn from new samples, as a result, gradually putting more weight on the board rather than on the tracking object.

4. Experiment

In this section, first, an ablation study of every module we propose above will be conducted to prove the function of every module. Then we will run the integrated tracking model on OTB100 [23] and a dataset collected by ourselves comparing with other advanced and popular tracking models to illustrate the advantage of our tracking method.

4.1 Ablation study of every module

The original tracking model is ECO-HC, and every module we propose above is expressed as shown in Table 1.

Table 1 Notation of every module

Speed-accuracy-balanced model	Constrained model updater	Motion model
Deep-HC	Renew	Kalman filter

Based on abundant experiments on OTB100, the proper size of the search area is about three or four times of the object's size. The ablation study result of every module added to the original tracking model is arranged in Table 2. The major difficult situations in tracking have been classified into 11 types: fast motion (FM), background clutter (BC), illumination variation (IV), motion blur (MB), deformation (DEF), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out-of-plane rotation (OPR), out-of-view (OV) and scale variation (SV).

Table 2 Ablation experiment result

Scene	Module	AUC	Scene	Module	AUC
MB	Deep-HC-renew	**0.626**	FM	Deep-HC-renew	**0.633**
	Deep-HC	0.617		Deep-HC	0.625
	Deep-HC-renew-KF	0.614		ECO-HC	0.621
	ECO-HC	0.610		Deep-HC-renew-KF	0.608
IPR	Deep-HC-renew-KF	**0.549**	OPR	Deep-HC-renew	**0.567**
	Deep-HC-renew	0.539		Deep-HC	0.556
	Deep-HC	0.532		ECO-HC	0.556
	ECO-HC	0.530		Deep-HC-renew-KF	0.552
BC	Deep-HC-renew-KF	**0.631**	LR	ECO-HC	**0.594**
	Deep-HC-renew	0.627		Deep-HC-renew-KF	**0.594**
	Deep-HC	0.611		Deep-HC-renew	0.592
	ECO-HC	0.583		Deep-HC	0.586
OCC	Deep-HC-renew-KF	**0.574**	IV	Deep-HC-renew-KF	**0.616**
	Deep-HC-renew	**0.574**		Deep-HC-renew	0.609
	Deep-HC	0.562		Deep-HC	0.595
	ECO-HC	0.562		ECO-HC	0.588
OV	Deep-HC-renew-KF	**0.548**	SV	Deep-HC-renew-KF	**0.594**
	Deep-HC-renew	0.530		ECO-HC	0.593
	ECO-HC	0.526		Deep-HC-renew	0.592
	Deep-HC	0.525		Deep-HC	0.587
	Deep-HC-renew	**0.615**		Deep-HC-renew	**0.568**
all	Deep-HC-renew-KF	0.612	DEF	Deep-HC	0.564
	Deep-HC	0.609		ECO-HC	0.558
	ECO-HC	0.604		Deep-HC-renew-KF	0.554

From the ablation study result, it is clear that speed-accuracy-balanced feature extraction module and constrained updater module could greatly enhance the performance of the original tracking model, especially in the motion blur (+2.62%), fast-moving (+4.11%), out-of-plane rotation (+1.98%) and deformation (+1.79%) scenes. Because comparing with the ECO-HC model which only simply uses FHOG and color features, Deep-HC-renew model utilizes additional high-level features obtained from convolution neural networks, therefore being better at handling variation in shallow features and consistency in complex deep features. However, different from those tracking models applying convolution neural networks through the whole tracking process, Deep-HC-renew model only applies convolution neural networks when the simple feature extraction model is not working well. Hence, while improving accuracy, this model reaches high speed at the same time. Moreover, the constrained updater ensures that some low-quality samples, like blur, partial occlusion and out-of-plane object would not be taken into training, as a result, it prevents the occurrence of degradation.

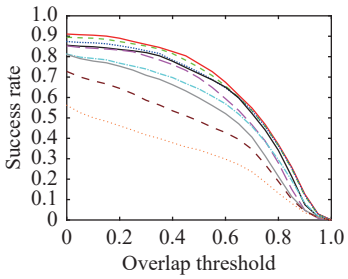
The Kalman filter based motion module would help the tracking model further improve its performance with almost no extra time spent, in background clutter (+8.23%), occlusion (+2.14%), and out-of-view (+4.18%) scenes. Because Klaman filter only adopts historical information before one frame and four simple motion

state variables (X, Y, V_X, V_Y), its data size is tiny and the calculation amount is also small. However, its prediction is reliable, particularly in those approximate linear regular motion scenarios. When an object moves extremely fast, its future location can be well predicted. Or when an object is occluded, though with no observation information, its location could also be estimated. Consequently, excellent performance is fully manifested in fast-moving and occluded situations.

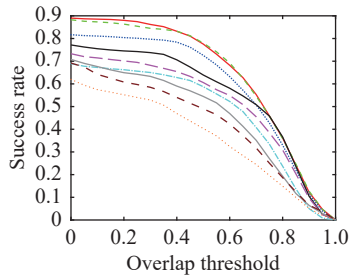
The reason why the original tracking model, ECO-HC, obtained good performance in low resolution and scale variation situations probably is that the object's features required for tracking in these situations are simple and just ordinary color and profile features are enough.

4.2 Integred model performance

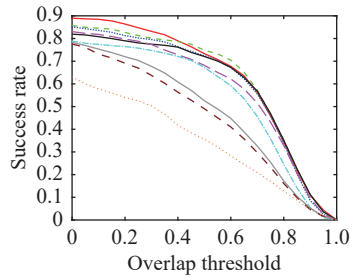
(i) Performance on OTB100: The experiment results are shown in Fig. 5, where the quantities of each sample are shown in the bracket. When the overlap score is greater than a given threshold, it is considered a successful frame of tracking. The percentage of successful frames is referred to as the success rate. By plotting the success rate against the threshold values, a success plot can be generated. The number in the bracket behind the subtitle means how many samples are classified into that kind of major difficult situations in tracking. From the experiment result shown in Fig. 5, we can learn that our tracking method has exceeded most of the popular and advanced tracking methods: CSK, KCF, LCT2, SiameseFC, staple, SRDCF, ECO-HC in tracking success rate at the whole OTB100 benchmark. In those different scenarios, like motion in blur scene, Deep-HC-renew derives the best performance and in occlusion, out-of-view, scale variation scenes Deep-HC-renew-KF gets the greatest results. Additionally, the speed of our method is also satisfying which is 21 fps, ran on Intel (R) Core (TM) i7-10700F CPU.



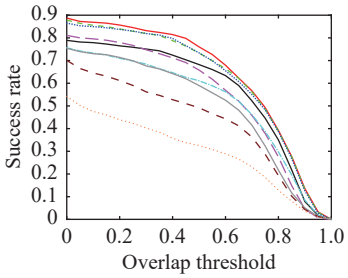
(a) Success plots of OPE-fast motion (43)



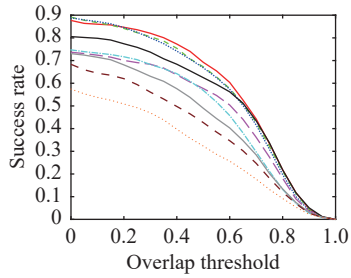
(b) Success plots of OPE-background clutter (30)



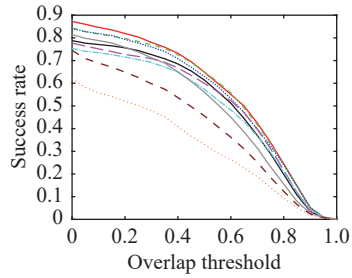
(c) Success plots of OPE-illumination variation (35)



(d) Success plots of OPE-motion blur (36)



(e) Success plots of OPE-deformation (44)



(f) Success plots of OPE-in-plane rotation (49)

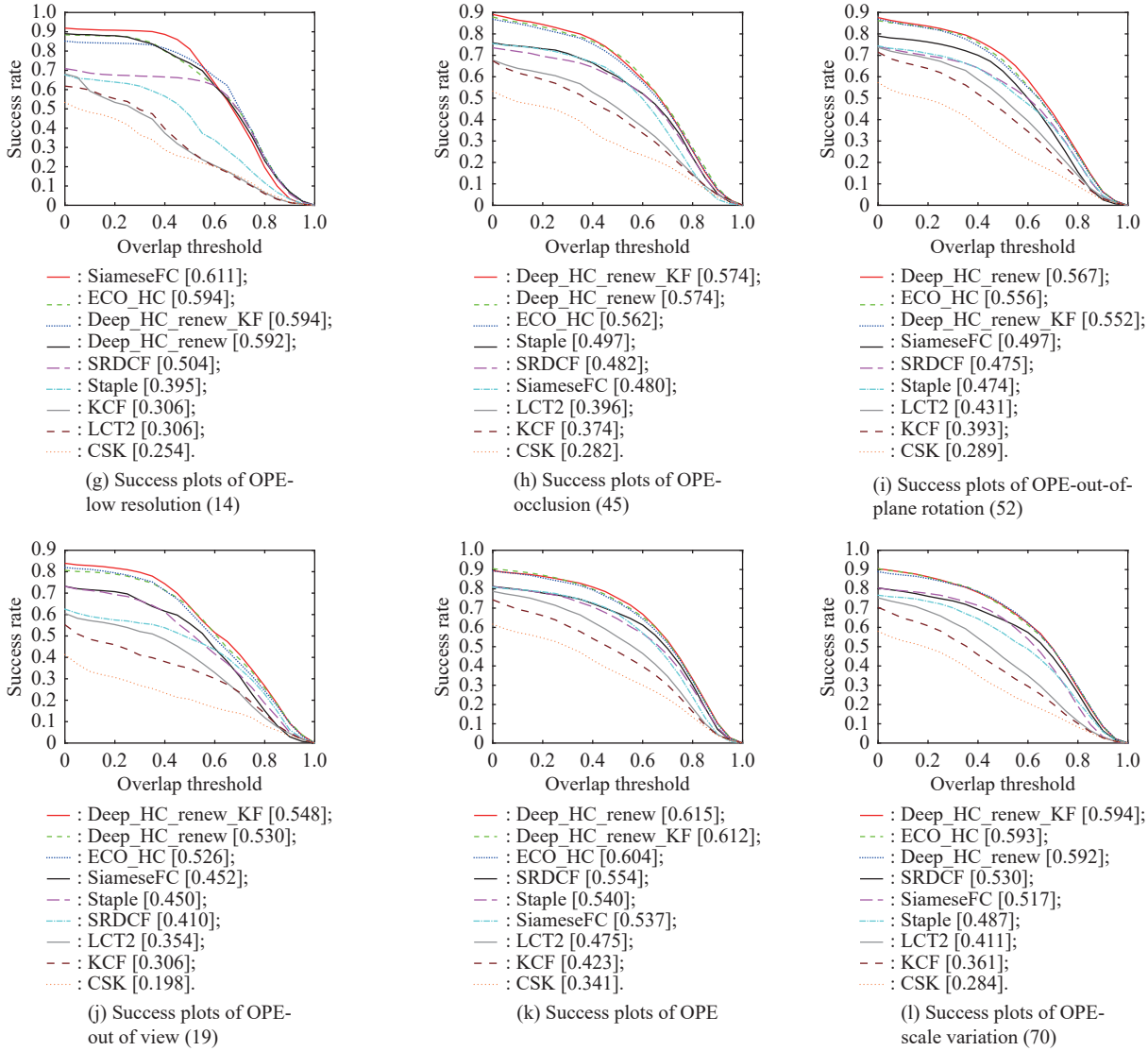


Fig. 5 Success plot of advanced tracking methods for 11 subsets on OTB100

Moreover, because our motion model is designed for those sceneries that an object moves in a regular pattern without sudden changes in the moving direction and speed, those datasets that satisfy this feature are picked out and listed as follows: Bolt, Human7, KiteSurf, Gym, Skiing, Surfer, Walking, Football1, Bird1, Bird2, Couple, Human3, Crowds, Football, MotorRolling and Jogging. They are all in the scene where the tracked object is doing regular exercise, like running, walking, surfing and flying. Our tracking method performs more outstanding on these occasions.

The result is shown in Fig. 6, for those appropriate regular movement situation, our proposed tracking approach exceeds ECO-HC 8.2% in AUC.

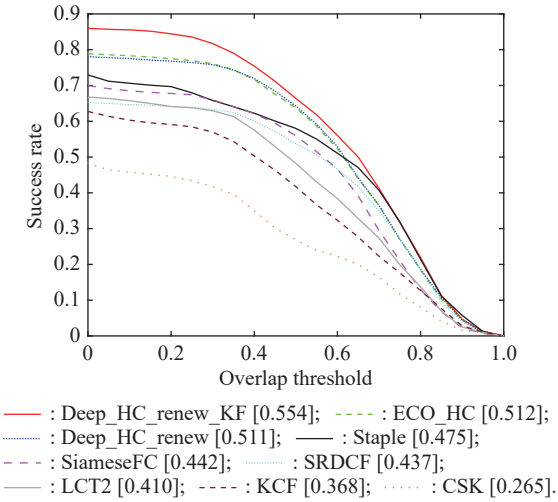
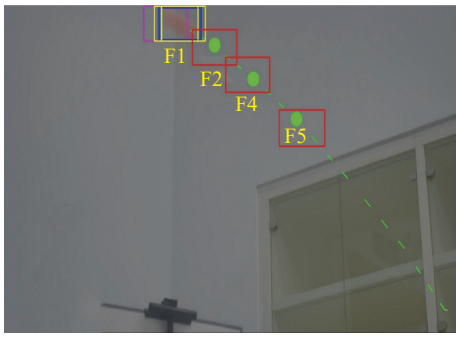


Fig. 6 Success plot of advanced tracking methods on regular movement dataset

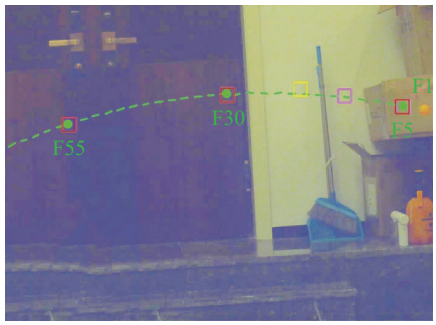
(ii) Performance on our own dataset: Due to the reason that OTB100 does not have enough datasets in regularly moving patterns, we collect our own dataset. We use high-speed camera (QianYanLang-2F01C) whose maximum resolution is 1 120×860, shooting frame rate is up to 20 000 fps (at a lower resolution), timestamp is accurate to 1/24 microseconds and supports synchronous shooting with multiple cameras, compatible with Nikon, Pentax, Canon and other DSLR lenses to collect 20 video segments in various scenarios of throwing PingPang balls, including different frame rates: from 30 fps to 100 fps, and different occasions: short-term occlusion, long-term occlusion and no occlusion.

The collected data in different scenes and track ing results obtained from different tracking methods are shown in Fig. 7−Fig. 9. The green dashed lines describe the trajectories of the PingPang, and those green dots represent the Ground Truth location of PingPang at Fi (the ith frame in the video segment). Moreover, the tracking result from different models at different Frame i is also marked as rectangular boxes in other colors.



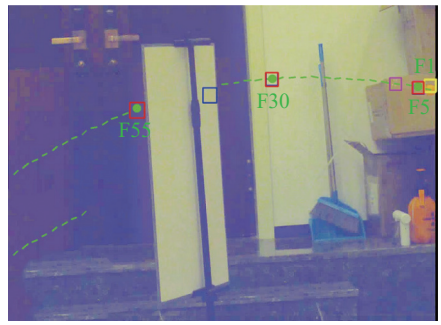
□ : OUR; □ : ECO-HC; □ : CSK; □ : KCF.

Fig. 7 Tracking result of different methods on 30 fps dataset



□ : OUR; □ : ECO-HC; □ : CSK; □ : KCF.

Fig. 8 Tracking result of different methods on 100 fps no occlusion dataset



□ : OUR; □ : ECO-HC; □ : CSK; □ : KCF.

Fig. 9 Tracking result of different methods on 100 fps short-term occlusion dataset

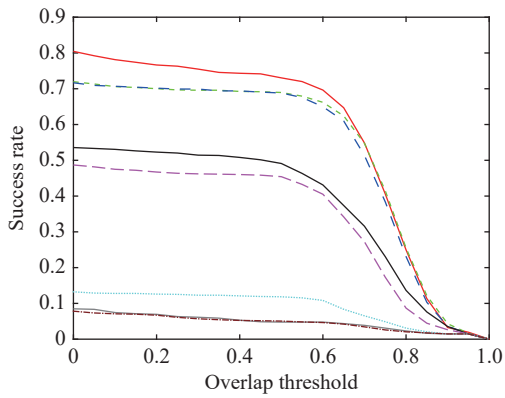
Fig. 7 shows a video segment collected from setting the frame rate of the camera to 30 fps. It is obvious that at such a low frame rate the object suffers severe motion blur and huge translocation. In such harsh situation, those normal tracking methods are impossible to catch up with the object and stagnated at the location of the first initialization. However, our approach has stable and accurate performance, keeping precisely locating the object.

Fig. 8 and Fig. 9 illustrate the tracking results of different methods on video segments obtained from setting camera's frame rate to 100 fps. In Fig. 8 and Fig. 9, the motion blur has been greatly alleviated for the high frame rate. Therefore, as shown in Fig. 8, ECO-HC and our method could accurately track the object from beginning to end. While, KCF and LCT2 could track the object at the initial frames, like in video frame 5, but miss the object later. This phenomenon demonstrates that our model could track object in complex and cluttered background more stably.

Additionally, when occlusion happens, without proper countermeasures, KCF, LCT2, CSK, ECO-HC lose the object inevitably. Our methods adopting Kalman filter based motion model perfectly predict the movement of the object when the observation state judgment model determines that it is occluded, tracking along its moving pattern without enlarging the search area and relocating the object when it appears again, like the tracking result of frame 55 as shown in Fig. 9.

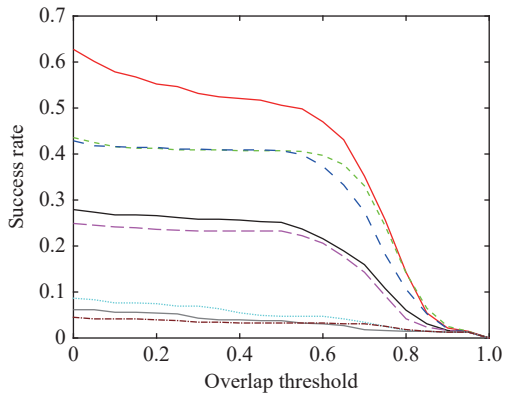
The experiment result is shown in Fig. 10, where the quantities of each sample are shown in the bracket. Our method improves the AUC of the original tracking model from 0.52 to 0.563, increasing 8%. In occlusion situation it improves 32%, and in fast motion scene it improves 8%. These results prove that our model by adding the observation state judgment module, speed-accuracy-balanced module, Kalman filter based motion module and constrained updater, enhances the tracking accuracy in

cluttered backgrounds and stability in fast-moving, occlusion scenes.



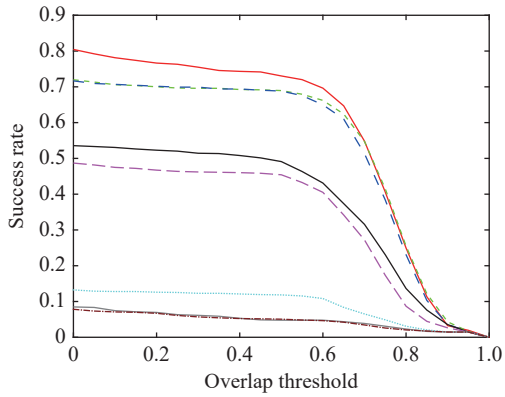
—: Deep_HC_renew_KF [0.563]; - - - -: Deep_HC_renew [0.527];
– –: ECO_HC [0.520]; ——: LCT2 [0.370]; – –: KCF [0.330];
·····: DFT [0.089]; ——: ORIA [0.047]; –·–·–: CSK [0.045].

(a) Success plots of OPE



—: Deep_HC_renew_KF [0.396]; - - - -: Deep_HC_renew [0.312];
– –: ECO_HC [0.300]; ——: LCT2 [0.187]; – –: KCF [0.169];
·····: ORIA [0.048];——: CSK [0.034]; –·–·–: DFT [0.030].

(b) Success plots of OPE - occlusion (7)



—: Deep_HC_renew_KF [0.563]; - - - -: Deep_HC_renew [0.527];
– –: ECO_HC [0.520]; ——: LCT2 [0.370]; – –: KCF [0.330];
·····: DFT [0.089]; ——: ORIA [0.047]; –·–·–: CSK [0.045].

(c) Success plots of OPE - fast motion (16)

Fig. 10 Tracking result of advanced methods on our own collected dataset

5. Conclusions

In this paper, given the limits of nowadays tracking methods, like the ignoring of handling the occlusion, difficulty in finding the balance between speed and accuracy, a new tracking method which takes tracking state observation, speed-accuracy balance, motion pattern and constrained update into consideration is proposed. This tracking method can run in real time 21 fps and perform pretty well in background clutter, occlusion, fast-moving, out-of-view scenarios. Abundant experiments on OTB100 and our own collected dataset thoroughly prove the efficiency of the proposed tracking model.

References

[1] KHATTAK A S, RAJA G, ANJUM N, et al. Integration of mean-shift and particle filter: a survey. Proc. of the 12th International Conference on Frontiers of Information Technology, 2014: 286−291.

[2] GHEDIA N, VITHALANI C, KOTHARI A M, et al. Existing research in video surveillance system. https://doi.org/10.1007/978-3-030-90910-9_2.

[3] ZHOU N, LAU L, BAI R, et al. A genetic optimization resampling based particle filtering algorithm for indoor target tracking. Remote Sensing, 2021, 13(1): 132–154.

[4] WAILA G S, KUMAR A, SAXENA A, et al. Robust object tracking with crow search optimized multi-cue particle filter. Pattern Analysis and Applications, 2020, 23: 1439–1455.

[5] ZHANG L. A survey of target tracking algorithms based on correlation filtering. International Core Journal of Engineering, 2022, 8(4): 566–576.

[6] DU S D, WANG S P. An overview of correlation-filter-based object tracking. IEEE Trans. on Computational Social Systems, 2021, 9(1): 18–31.

[7] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 37(3): 583–596.

[8] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters. Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010: 2544−2550.

[9] BERTINETTO L, VALMADRE J, GOLODETZ S, et al. Staple: complementary learners for real-time tracking. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 1401−1409.

[10] MA C, YANG X K, ZHANG C Y, et al. Long-term correlation tracking. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 5388–5396.

[11] DANELLJAN M, HAGER G, SHAHBAZ K, et al. Learning spatially regularized correlation filters for visual tracking. Proc. of the IEEE International Conference on Computer Vision, 2015: 4310−4318.

[12] DANELLJAN M, BHAT G, SHAHBAZ K, et al. Eco: efficient convolution operators for tracking. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6638−6646.

[13] LIU S, LIU D Y, SRIVASTAVA G, et al. Overview and methods of correlation filter algorithms in object tracking. Complex & Intelligent Systems, 2021, 7: 1895–1917.

[14] ZHAO S K, SUN K W, JI Y F, et al. Correlation filter-based object tracking algorithms. Proc. of the IEEE International Conference on Information Communication and Signal Processing, 2020: 57−62
[15] WANG D C, BAI C S, WU K J. Survey of video object detection based on deep learning. Journal of Frontiers of Computer Science & Technology, 2021, 15(9): 1563–1578. (in Chinese)
[16] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-convolutional siamese networks for object tracking. Proc. of the Computer Vision-ECCV 2016 Workshops, 2016: 850−865.
[17] VOIGTLAENDER P, LUITEN J, TORR P H S, et al. Siam R-CNN: visual tracking by re-detection. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 6578−6588.
[18] LI B, WU W, WANG Q, et al. Siamrpn++: evolution of siamese visual tracking with very deep networks. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 4282−4291.
[19] WANG N, ZHOU W G, WANG J, et al. Transformer meets tracker: exploiting temporal context for robust visual tracking. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 1571−1580.
[20] DANELLJAN M, BHAT G, KHAN F S, et al. Atom: accurate tracking by overlap maximization. Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 4660−4669.
[21] BHAT G, DANELLJAN M, GOOL L V, et al. Learning discriminative model prediction for tracking. Proc. of the IEEE/CVF International Conference on Computer Vision, 2019: 6182−6191.
[22] WU Y, LIM J, YANG M H. Online object tracking: a benchmark. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2411−2418.
[23] WU Y, LIM J, YANG M H. Object tracking benchmark. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834–1848.
[24] KRISTAN M, MATAS J, LEONARDIS A, et al. The ninth visual object tracking vot2021 challenge results. Proc. of the IEEE/CVF International Conference on Computer Vision, 2021: 2711−2738.
[25] MUELLER M, SMITH N, GHANEM B. A benchmark and simulator for uav tracking. Proc. of the Computer Vision-ECCV 2016 Workshops, 2016: 445−461.
[26] JIANG M, LI R, LIU Q S, et al. High speed long-term visual object tracking algorithm for real robot systems. Neurocomputing, 2021, 434: 268–284.
[27] MA C, HUANG J B, YANG X, et al. Adaptive correlation filters with long-term and short-term memory for object tracking. International Journal of Computer Vision, 2018, 126: 771–796.
[28] YANG Z F, CHEN X, Optimized searching strategy for kcf object tracking algorithm. Journal of Wuhan Institute of Technology, 2019, 41(1): 98−102. (in Chinese)
[29] DANELLJAN M, ROBINSON A, SHAHBAZ K F, et al. Beyond correlation filters: learning continuous convolution operators for visual tracking. Proc. of the Computer Vision-ECCV 2016 Workshops, 2016: 472−488.
[30] LI X, ZHOU J L, HOU J Q, et al. Research on improved moving object tracking method based on eco-hc. Journal of Nanjing University (Natural Science), 2020, 56(2): 216–226. (in Chinese)
[31] WANG N Y, SHI J P, YEUNG D Y, et al. Understanding and diagnosing visual tracking systems. Proc. of the IEEE International Conference on Computer Vision, 2015: 3101−3109.

Biographies

LI Yuran was born in 2000. She received her B.E. degree in automation from Xi'an Jiaotong University, Xi'an, China, in 2021. She is currently pursuing her M.E. degree in control science and engineering at Shanghai Jiaotong University, Shanghai, China. Her current research interests include computer vision and deep learning.

E-mail: liyuran20000220@sjtu.edu.cn

LI Yichen was born in 1993. He received his B.S. degree in detection, guidance and control technology from Northwestern Polytechnical University, Xi'an, China, in 2016 and Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2022. He is now a postdoc in control science and engineering with Shanghai Jiao Tong University, Shanghai, China. His current research interests include underwater multi-robot localization and trajectory planning, wireless networks, and information fusion.

E-mail: liyichensjtu@sjtu.edu.cn

ZHANG Monan was born in 1993. He received his B.E. degree in measurement and control technology and instrumentation from Harbin Engineering University, Harbin, China, in 2016, and M.E. degree in aeronautical and astronautical science and technology from Harbin Institute of Technology, Harbin, China, in 2018. He is currently pursuing his Ph.D. degree in control science and engineering at Shanghai Jiao Tong University, Shanghai, China. His current research interests include target detection and tracking in computer vision.

E-mail: mnzhang@sjtu.edu.cn

YU Wenbin Was born in 1983. He received his Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2016. He is currently an associate professor with the Department of Automation, Shanghai Jiao Tong University. His research interests include data fusion and control strategy for AUV system.

E-mail: yuwenbin@sjtu.edu.cn

GUAN Xinping was born in 1963. He received his Ph.D. degree in control and systems from Harbin Institute of Technology, Harbin, China in 1999. In 2007, he joined the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Distinguished University Professor and the Director of the Key Laboratory of Systems Control and Information Processing, Ministry of Education of China. His current research interests include wireless sensor networks, ground-air communication of aircraft, and cognitive radio networks and their applications in industry.

E-mail: xpguan@sjtu.edu.cn