Neural Computing and Applications (2021) 33:2229–2248
https://doi.org/10.1007/s00521-020-05067-3

ORIGINAL ARTICLE

Performance evaluation of low resolution visual tracking for unmanned aerial vehicles

Yong Wang[1] · Xian Wei[2] · Hao Shen[3] · Jilin Hu[4] · Lingkun Luo[5]

Received: 26 September 2019 / Accepted: 3 June 2020 / Published online: 27 June 2020

Abstract
Several datasets for unmanned aerial vehicle (UAV) visual tracking research have been released in recent years. Despite their usefulness, whether they are sufficient for understanding the strengths and weakness of different resolution videos tracking remains questionable. Tracking in low resolution videos is a critical problem in UAV tracking. To address this issue, we construct a group of low resolution tracking datasets and study the performance of different trackers on these datasets. We find that some trackers suffered more performance degradation than others, which brings to light a previously unexplored aspect of the tracking methods. The relative rank of these trackers based on their tracking results on the datasets may change in the presence of low resolution. Based on these findings, we develop a multiple feature tracking framework which takes advantage of image enhancement scheme to improve image quality. In addition, we utilize the forward and backward tracking to evaluate multiple feature tracking results. Experimental results demonstrate that our tracker is competitive in performance to state-of-the-art methods in different resolutions scenarios. We believe our studies can provide a solid baseline when conducting experiments for low resolution UAV tracking research.

Keywords UAV tracking · Low resolution dataset · Fusion algorithm · Image enhancement

1 Introduction

The wide spread of the usage of commercial unmanned aerial vehicle (UAV) makes the UAV tracking being attracted more and more attentions. There are four UAV tracking datasets released in recent years [1–4] which greatly promote the study of UAV tracking. Though many tracking algorithms are proposed, there still many problems exist, especially related to the characteristic of UAV. For example, the low resolution problem is a key problem in UAV tracking.

Most of the time, targets in UAV videos are in low resolution due to the flight attitude. This degrades the performance of UAV tracking. There are many differences in features between big targets and small targets as showed in vehicle detection [5], and pedestrian detection [6]. Small targets make the features ambiguous to the background. Although there are many works in the literature on visual object tracking, few works study low resolution tracking. Figure 1 demonstrates the differences of response map in two different resolutions.

There are mainly two approaches for low resolution tracking [7, 8]. The first one is using metric learning to enhance the discriminability of the tracker [7]. However, there is an offline training procedure which requires collecting training data.

Another one [8] is using super resolution technique to restore the target first. And then ECO method [9] is used to track the restored videos. This method has two flaws. Firstly, it separates the low resolution tracking procedure into two independent parts, super resolution and tracking.

✉ Xian Wei
xian.wei@fjirsm.ac.cn

1 School of Aeronautics and Astronautics, Sun Yat-Sen University, Guangzhou, China

2 Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, Fuzhou, China

3 Fortiss GmbH, Technical University of Munich, Munich, Germany

4 Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates

5 School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai, China

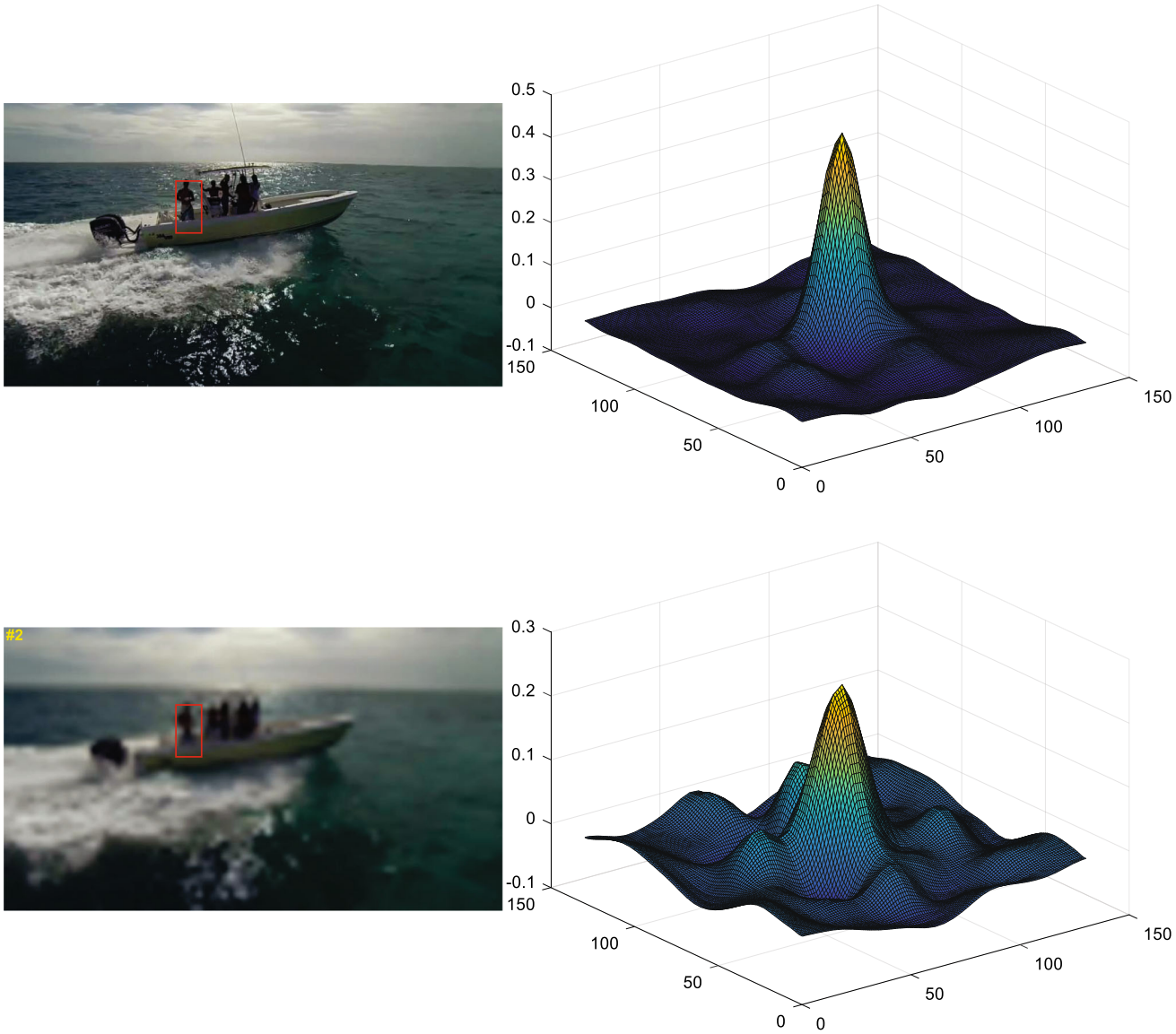

Fig. 1 Demonstration of the target response in different resolution. In the first row, the video resolution is 1280 × 720. The response map is sharp. In the second row, the video resolution is 80 × 45. There is only a contour of the target. Many noises exist in the response map

Secondly, super resolution is a difficult problem in image processing. Though the deep learning methods enhance the super resolution technique, training a deep learning model is computational expensive. Tracking in different low resolutions is investigated in [10]. Four videos are changed with different resolutions and ten tracking methods are compared.

In this paper, we give a thorough study of UAV tracking in different low resolutions. We mainly focus on three aspects. Firstly, we adopt the DTB70 dataset as our test bed. The videos are resized from the second frame to the last frame. The first frame is used to train the model accurately. Previous work only uses a small number of videos (four and eight videos used in [10] and [8], respectively). Large number of videos gives statistical results in low resolution tracking. These results enable us to give a more objective conclusion. In addition, recent state-of-the-art tracking methods are tested on these low resolution datasets.

Secondly, histogram of oriented gradient (HOG) [11] and color name [36] feature are utilized. In the low resolution videos, target shape is an important clue as shown in Fig. 1. HOG feature is robust to low resolution and illumination variation. Color name feature is complementary to HOG as it is robust to deformation and sensitive to illumination variation. The deep learning features have achieved appealing results in visual tracking [13]. However, deep learning features are sensitive to the size of the

target. The low resolution target will degrade the performance of visual tracking methods. In addition, deep learning features are computational expensive. They are not suitable in a UAV tracking platform. Furthermore, we develop a robust fusion algorithm to adaptively combine the tracking results.

Thirdly, a higher resolution image can generate a sharper response than a lower resolution image (as shown in Fig. 1). We adopt an image enhancement algorithm to improve the quality of our videos. The first frame can be used as an initial training data. Since the differences between two consecutive frames are small, the previous frame with details enhanced can be used as a guidance image for the next frame. Thus the qualities of frames can be improved sequentially.

In summary, our contributions are three-fold.

Firstly, we generate a group of UAV low resolution tracking datasets. And recent state-of-the-art tracking methods are tested on these datasets. To the best of our knowledge, this is the first group of tracking datasets to systematically study low resolution tracking.

Secondly, we adopt image enhancement algorithm to improve the quality of the low resolution tracking datasets and improve tracking results.

Thirdly, we develop a multiple features fusion algorithm to test on the UAV low resolution datasets and achieves appealing results.

2 Related work

There are many literatures on visual tracking. For a thorough review, readers can refer to [18, 19]. In this section, we only review the methods that most related to our work.

2.1 Low resolution tracking

In [7], the authors propose an approach which is based on discriminative metric preservation. It can preserve the data affinity structure in the high resolution feature space for effective and efficient matching of low resolution images. A super-resolution tracker is presented in [8]. This method integrates a super-resolution reconstruction algorithm into correlation filter tracking framework to enhance the resolution of the object. Seven correlation filter based trackers are compared and tested on eight low resolution sequences selected from the OTB dataset [19].

Nine different low resolutions on four videos are created in [10]. The resolution of the videos is reduced from the original size to the one-ninth of the original size. Ten algorithms are tested on these sequences. The experiments demonstrate that the performance of some methods is not stable on these video sequences with different resolutions.

The structured SVM trackers [40, 41] have achieved appealing performance in low-resolution tracking. And low resolution situations are studied in RGB-T tracking [42]. Too little work has been devoted to low resolution tracking. Thus, we systematically investigate low resolution tracking in this paper and comparison results are provided. We hope our work can bridge the gap between visual tracking and low resolution problem.

2.2 Correlation filter tracking

Recently, correlation filter based tracking methods have become popular. Initially, in [27], the authors develop a Minimum Output Sum of Squared Error (MOSSE) filter with an adaptive online training strategy. This is a fast and robust tracking method. It only uses intensity feature which is a single feature channel. Then multiple kernel features are developed in kernelized correlation filter (KCF) method [28] which utilizes the circulant structure to achieve fast tracking through the Fourier transformation. That is, a classifier is trained by using kernel regularized least squares with dense sampling around the predicted object position.

In [29] the correlation filter based tracking framework is incorporated with adaptive scale estimation. HOG [11] and color features are incorporated in Staple [30]. To alleviate the problem of corrupted training samples in correlation filter tracking framework, the qualities of the samples are dynamically measured [31]. This scheme enables contaminated samples to be down-weighted and increases the weights of correct ones.

To improve performance in long term tracking, a sparse coding detector is developed in the correlation filter tracking framework [35]. In order to use context information, a spatial regularization term is considered in correlation tracking framework [32]. High dimensional features are reduced and an adaptive model update algorithm is developed to improve tracking speed and results [9]. Background information is taken into consideration in [34] and alternating direction method of multipliers (ADMM) algorithm is employed to accelerate the computation procedure. This work is extended to [26] by combining hard negative example learning technique. Spatial temporal information is taken into consideration in [25] to improve tracking performance.

Hierarchical deep features methods with correlation filter tracking framework are exploited in [12, 13]. Multi-expert collaboration tracker with hierarchical deep features is built in [24]. Forward and backward tracking scheme is added in the correlation filter tracking to evaluate tracking quality [23].

2.3 Tracking datasets

There are several widely used object tracking datasets, e.g., OTB [19], VOT [14, 20], ALOV [18], LaSOT [22], and TC-128 [21]. These datasets are mainly for generic object tracking. The VIVD dataset [33] is an early UAV tracking dataset, but there are only nine videos. The UAV123 [1], DTB70 [2], UAVDT [3] and VisDrone [4] are released in recent years. There are 123, 70, 50 and 60 videos in the UAV123, DTB70, UAVDT and VisDrone, respectively. The evaluation metrics are precision score and success score.

In this work, we choose the DTB70 as our test bed. Four different low resolution UAV tracking datasets are generated. The BACF [34], Staple [30], ECO [9], SRDCFdecon [31], HDT [24], STRCF [25], deepRedetection [35] and fDSST [29] are tested on the datasets. The ECO and BACF are also used in the UAVDT and VisDrone as benchmark results. To deal with feature ambiguous in low resolution sequences, HOG and color names features are employed to track in correlation filter framework. A fusion algorithm is proposed to effectively and robustly fuse the tracking results. In addition, an image enhance algorithm is employed to improve the quality of low resolution images.

3 Tracking algorithm

The workflow of our tracking algorithm is illustrated in Fig. 2. Our tracking is based on the KCF framework. More specifically, multiple feature extraction is first carried out. Next, the quality of the low resolution image is enhanced and forward tracking is implemented. Then the previous image is also enhanced and the backward tracking is carried out. The fusion algorithm is designed to combine these tracking results. The final position is obtained and the model updating is carried out.

3.1 Multiple features

Deep learning features have achieved appealing performance in recent years. In UAV tracking, the speed has been an important issue besides its performance. Unfortunately, Deep learning based tracking methods are computation expensive. For example, the HDT [24] is 1 frame per second (fps). On the contrary, HOG feature based methods [9, 25, 34] are more than 10 fps. In UAV situations, the power is limited. Therefore, we only choose the three handcrafted features to balance the accuracy and speed.

In our tracking framework, three features are adopted, HOG, color names and concatenated of HOG and color names (fusion feature). HOG feature is robust to illumination changes and scale variations. Color feature is effective to deformation. In addition, concatenated of HOG and color features are employed to improve feature description. A single feature along can not deal with all challenge situations. HOG and color feature can fail in some scenarios. We observe that if HOG feature fail, color feature is effective. Meanwhile, if color feature fail, HOG feature is effective. Therefore, we use three strategies to improve feature description. Firstly, we use multiple features for low resolution tracking. These features are complementary to each other. Next, to obtain better object features, guided filter is utilized to improve image quality. Moreover, to prevent tracking lost, forward and backward tracking scheme is employed to evaluate tracking results. This scheme provides tracking reliability of each feature.

3.2 Kernelized correlation filter tracker

In this section, we first give a brief introduction to the KCF method. Readers can refer to [28] for more details. The KCF classifier is trained on an image patch x with the size of $W \times H$. The training patches can be generated based on the center of previous tracked position. The correlation filter is able to cover all possible training candidates $x_{w,h}, (w,h) \in \{0,\cdots,W-1\} \times \{0,\cdots,H-1\}$ through the cyclic property and padding. $y(w, h)$ is the label of $x_{w,h}$. It can be modeled via a Gaussian function. The value of a

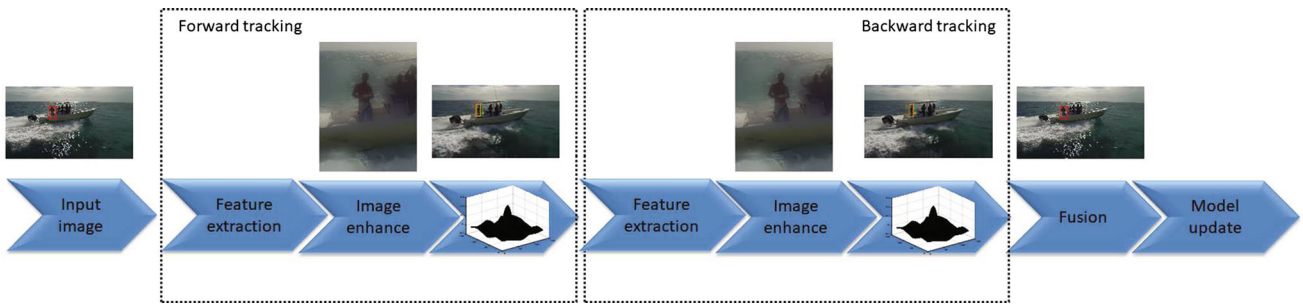

Fig. 2 Workflow of our tracking algorithm

centered target is set to 1, and the values around the target smoothly decay to 0. Function $f(z) = w^T z$ is to minimize the squared error of samples $x_{w,h}$ and the label $y(w, h)$. The function can be computed as follows,

$$min_{\omega} \sum_{w,h} |<\phi(x_{w,h}, \omega)> - y(w,h)| + \lambda||\omega||^2, \tag{1}$$

where operator $<>$ is inner product, ϕ is the mapping function, *lambda* is a parameter for the regularization term. The w can be represented as $\omega = \sum_{w,h} \alpha(w,h)\phi(x_{w,h})$,

$$\alpha = \mathcal{F}^{-1}(\frac{\mathcal{F}(y)}{\mathcal{F}(k^x) + \lambda}), \tag{2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier Transform and inverse Fourier Transform, respectively, $k^x = k(x_{w,h}, x)$,

The target appearance $\hat{x}$ can be updated during tracking. The models of correlation filter tracker include the trained target appearance $\hat{x}$ and the coefficients $\mathcal{F}(\alpha)$. During online tracking, a new patch z is cropped and the confidence score is calculated according to the following equation,

$$f(z) = \mathcal{F}^{-1}(\mathcal{F}(k^z) \odot \mathcal{F}(\alpha)), \tag{3}$$

where $\odot$ is the element-wise product.

3.3 Image enhancement

Guided filter algorithms [15–17] are proposed to improve image quality. In these algorithms, a guided filter uses a guidance image to boost the detail information and suppress the noise. In this work, the first frame is used as an initial guidance image. The second image is firstly enhanced via the previous image and then can be used as a guidance image for the third frame. And the procedure is carried out sequentially. In [16], the method achieves appealing performance and in fast speed which is suitable for UAV tracking. Therefore, we adopt the guided filter [16] in our work. Here, we give a brief description of the guided filter [16].

I is a guided image, w_k is a window centered at the pixel k. q is the filter output through a linear transform,

$$q_i = a_k I + b_k, i \in w_k, \tag{4}$$

where (a_k, b_k) are two constants values in the window, i and k are the indices. The average of local variances for all pixels is defined as,

$$\Gamma = \bar{\sigma}^2 = \frac{1}{N}\sum_{k=1}^{N} \sigma_k^2, \tag{5}$$

where σ_k^2 is the local variance of I in w_k. The values (a_k, b_k) in Eq. (4) can be obtained by minimizing the following function,

$$E(a_k, b_k) = min \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \lambda \Gamma a_k^2), \tag{6}$$

where p is the input image. The values a_k and b_k can be computed as follows,

$$a_k = \frac{1}{1 + \lambda \frac{\Gamma}{\sigma_k^2}}, \tag{7}$$

$$b_k = \bar{p_k} + a_k \mu_k, \tag{8}$$

The final filtered output q is calculated as,

$$q_i = \frac{1}{|w|}\sum_{i \in w_k} (a_k I_i + b_k) = \bar{a_i} I_i + \bar{b_i}, \tag{9}$$

where $\bar{a_i} = \frac{|w|}{\sum_{k \in w_k} a_k}$ and $\bar{b_i} = \frac{|w|}{\sum_{k \in w_k} b_k}$.

The details enhancement can be achieved as follows,

$$Z_{enh}(p) = X(p) + \theta e(p), \tag{10}$$

where $X(p)$ is the filtered output, e is the differences between the original image and filtered output, θ is a constant parameter. The enhanced results are illustrated in Fig. 3. The first row is the results of second frame. The second row is the results of the 50th frame. The left column is the original low resolution images. The right column is the enhanced results.

3.4 Forward and backward tracking

Figure 4a illustrates the forward and backward tracking scheme in the original DTB70 dataset. The forward tracking utilizes the correlation filter tracking framework as described in Sect. 3.1 with three different features. The backward tracking is the same to the forward but in a time inverse order. That is, from frame t+1 to frame t. the green, yellow and black rectangles in frame t+1 are tracked positions of HOG, color names and fusion feature, respectively. The same color rectangles are the backward tracking results of the corresponding features.

Figure 4b illustrates the forward and backward tracking scheme in the low resolution dataset. The cropped subwinow is first enhanced via previous subwindow. The correlation filter tracker is implemented on the enhanced subwindow.

In the backward procedure, the enhanced subwindow S_{en} is first treated as a guidance image and used to enhance the low resolution subwindow in frame t. and then the correlation filter tracker is implemented on the enhanced sub-

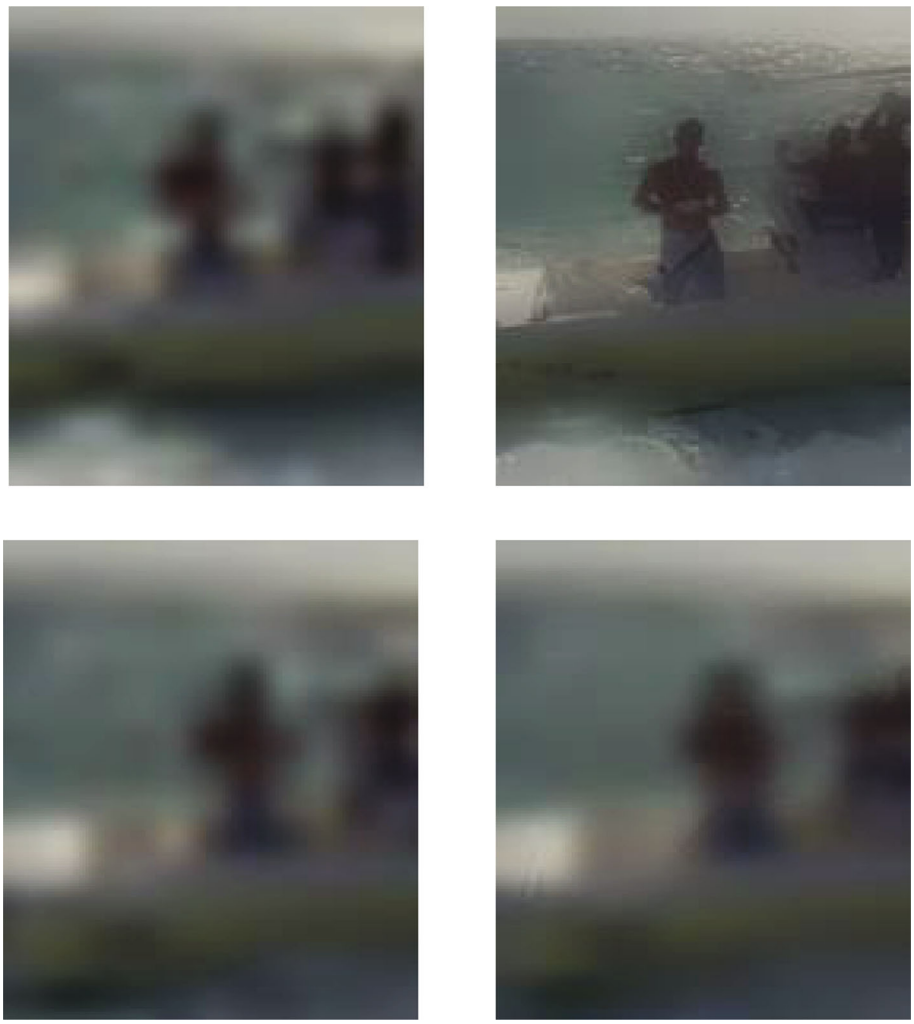

Fig. 3 Demonstration of image enhanced results. The first row is the results of second frame. The second row is the results of the 50th frame. The left column is the original low resolution images. The right column is the enhanced results

window. Meanwhile, the S_{en} is saved as a guidance image for the frame $t+2$ in the forward tracking procedure.

3.5 Fusion algorithm

The overlap and distance error are two metrics used to measure the quality of tracking results. We utilize these two metrics to evaluate the differences between forward and backward tracking. In our observation, the differences between forward and backward tracking are an indicator of the quality of the tracking results [23]. Here, we consider the tracking differences between different features in both forward and backward step. In the forward tracking step, overlap error and distance error can be computed as follows,

$$Overlap_f = \frac{area_{f_i^t} \cap area_{f_j^t}}{area_{f_i^t} \cup area_{f_j^t}}, \tag{11}$$

$$Dis_f = \sqrt{(x_{f_i^t} - x_{f_j^t})^2 + (y_{f_i^t} - y_{f_j^t})^2}, \tag{12}$$

where $area_{f_i^t}$ and $area_{f_j^t}$ are the forward tracked area of the ith and jth feature in frame t. $(x_{f_i^t}, y_{f_i^t})$ represent the forward tracked center of the ith feature in frame t. In the backward tracking step, the difference between the result of feature and the tracked position of last frame is also taken into consideration. Similarly, the overlap error $Overlap_b$ and distance error Dis_b can be computed following Eqs. (11) and (12), respectively.

The average and variance of the overlap are computed according to the results of forward and backward steps,

$$Ave_o = \sum (Overlap_f + Overlap_b)/3, \tag{13}$$

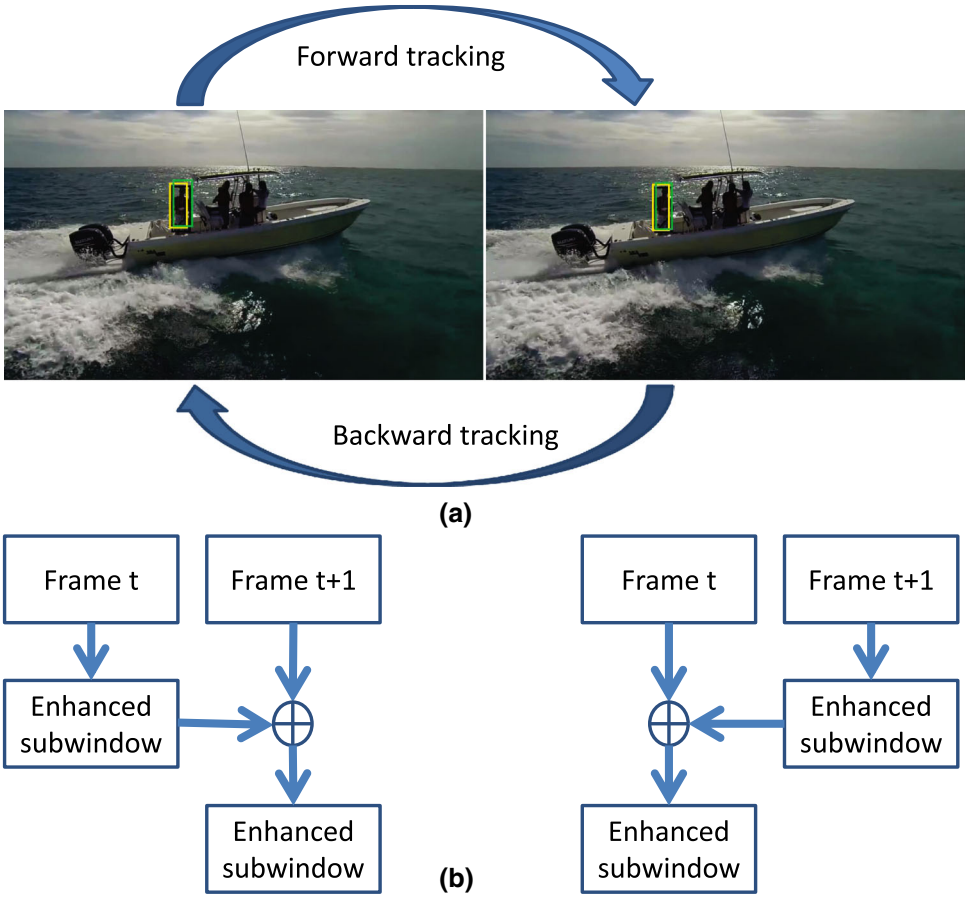

Fig. 4 Illustration of forward and backward tracking procedure

$$Var_o = \sqrt{\frac{1}{3}\sum_{1}^{3}(Overlap_f - Ave_o)}, \tag{14}$$

Similarly, the average and variance of the distance are computed as follows,

$$Ave_d = \sum(Dis_f + Dis_b)/3, \tag{15}$$

$$Var_d = \sqrt{\frac{1}{3}\sum_{1}^{3}(Dis_f - Ave_d)}, \tag{16}$$

The weights of overlap and distance errors is computed as follows,

$$W_o = Ave_o/(Var_o + \theta), \tag{17}$$

$$W_d = Ave_d/(Var_d + \theta), \tag{18}$$

where θ is a small value. The final weight is the fusion of the weights of overlap and distance,

$$W_i = \lambda \times W_o + (1 - \lambda) \times W_d, \tag{19}$$

where λ is a parameter.

The final position in frame $t + 1$ can be computed as follows,

$$z_{cf} = \beta_1 \times z_1 + \beta_2 \times z_2 + \beta_3 \times z_3, \tag{20}$$

where β_1, β_2 and β_3 are the normalized weight of W_1, W_2 and W_3 , respectively, z_1, z_2 and z_3 are the three forward tracking results.

Finally, the model is updated adaptively.

$$\mathcal{F}(\alpha)_i^t = \begin{cases} (1-\beta)\mathcal{F}(\alpha)_i^{t-1} + \beta\mathcal{F}(\alpha)_i, & if \quad w_i^t > threshold \\ \mathcal{F}(\alpha)_i^{t-1}, & else \end{cases} \tag{21}$$

$$x_i^t = \begin{cases} (1-\beta)x_i^{t-1} + \beta x_i, & if \quad w_i^t > threshold \\ x_i^{t-1}, & else \end{cases} \tag{22}$$

where β is a parameter, threshold is pre-defined.

Figure 5 illustrates our tracking algorithm.

Fig. 5 Our tracking algorithm

Algorithm: tracking algorithm
Input: t-th frame F_t, previous tracking state s_{t-1}. Output: Find the target location s_t.
1. Input the first frame, initial position. 2. Train the model. Subwindow is saved. 3. While video not end 4. %region of interest. 5. Crop subwindow. 6. For each feature 7. %HOG, color, HOG & color 8. Feature extraction. 9. % Forward tracking 10. Image enhancement according to equation (10). 11. Forward correlation tracking according to equation (3). 12. % Backward tracking 13. Backward image enhancement according to equation (10). 14. Backward correlation tracking according to equation (3). 15. %Adaptive weight 16. Overlap computed according to equation (11). 17. Distance computed according to equation (12). 18. Weight computed according to equation (19). 19. End 20. %Adaptive fusion 21. Position computed according to equation (20). 22. Subwindow is saved. 23. %Model update scheme. 24. If >Threshold 25. Model update according to equation (21). 26. End 27. End

4 Experiment

4.1 Experiment setup

4.1.1 Datasets

Comprehensive study has been performed on all the test videos in the DTB70 [2], which consists of 70 videos and covering eleven different tracking scenarios. They are aspect ratio variation (ARV), scale variation (SV), deformation (DEF), occlusion (OCC), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), fast camera motion (FCM), similar objects around (SOA), background cluttered (BC), and motion blur (MB). This dataset covers different types of camera motion including both translation and rotation. The target types are humans, animals and rigid objects. All the sequences capture outdoor scenes.

Four different low resolution versions of the DTB70 are generated. The resolution of the videos is reduced with the width and height by increasing levels with varying parameters 2, 4, 8, 16. The original resolution of the sequences is 1280 × 720. The resolution of the generated sequences is 640 × 360, 320 × 180, 160 × 90, and 80 × 45, respectively. They are denoted as LR1, LR2, LR3 and LR4 datasets, respectively.

Figure 6 (a) shows a frame from each dataset. Figure 6 (b) and (c) show the width and height distributions of the ground-truth bounding boxes over all sequences. The width in original sequences is mainly around 80 pixels, while in the LR4 dataset it reduces to 5 pixels. This represents the challenge of the LR4 dataset since small target is a difficult problem in visual tracking and detection.

4.1.2 Compared trackers

The BACF [34], Staple [30], ECO [9], SRDCFdecon [31], HDT [24], STRCF [25], deepRedetection [35] and fDSST [29] are tested on the datasets. These methods are all correlation filter based trackers and achieve better tracking performance than previous methods such as sparse representation based [37, 38] and multi-task based [39].

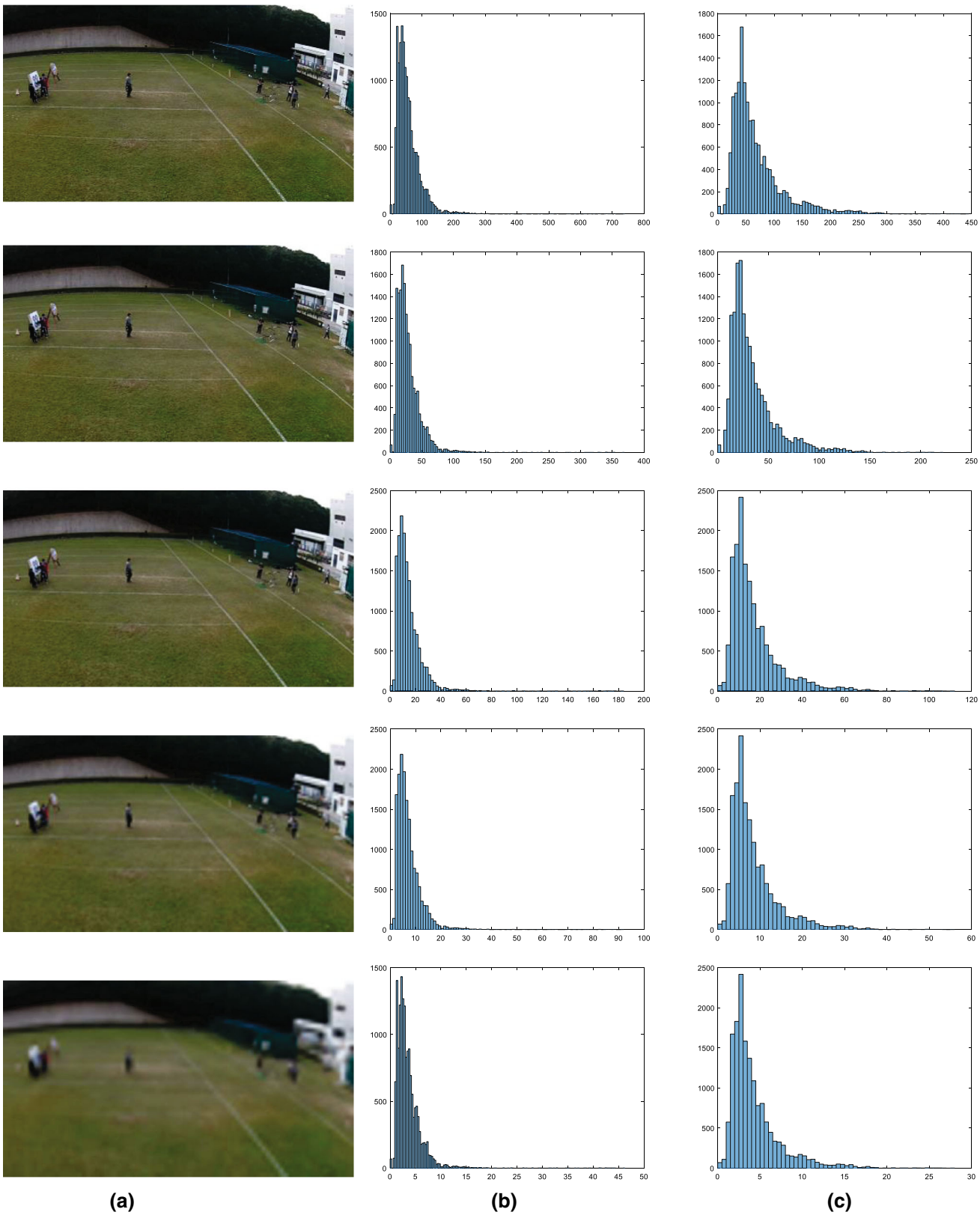

Fig. 6 Distribution of ground-truth bounding box size over all sequences. The first row is the original sequence with resolution of 1280 × 720. The second row is the sequence with resolution of 640 × 360. The third row is the sequence with resolution of 320 × 180. The fourth row is the sequence with resolution of 160 × 90. The last row is the sequence with resolution of 80 × 45

4.2 Evaluation metrics

One Pass Evaluation (OPE) is adopted to test the performance of these trackers. The OPE evaluation runs tracking methods only once on a video. Precision and success scores are generated to analyses the performance of these methods. The precision score is defined as the Euclidean distance between the predicted centers and ground truth centers. Percentage of frames is defined as predicted location lies within a pre-defined threshold distance from the ground truth. The threshold is 20 pixels.

The success score is the overlap between the predicted bounding box and ground truth bounding box. The overlap score is computed as,

$$o_s = \frac{r_t \cap r_g}{r_t \cup r_g}, \tag{23}$$

where r_t and r_g are the predicted bounding box and ground truth bounding box, respectively. $\cap$ and $\cup$ represent the intersection and union of two regions, respectively. If the score is greater than a threshold, the frames are referred to successful frames. The threshold is 0.5.

4.3 DTB70 dataset

Figure 7 shows overall performance of the ten trackers in terms of success and precision plots in the DTB70 dataset. Our method ranks the third on the success score of all OPE. In the success plot, the proposed method achieves the AUC of 0.403.

The ECO tracker, STRCF tracker and the deepRedetection achieve the first, second and third in the success score and precision score, respectively. In ECO, factorized convolution operator is used to improved HOG feature. Moreover, generative sample space model is developed to improve training samples. In addition, there are model update scheme which uses Gaussian mixture model. In STRCF, there are spatial and temporal constraints in the correlation filter. And ADMM is employed. The deepRedetection method uses deep features and a detector to re-detect target if tracking failure occurs. In addition, the HDT method is based on deep learning, which leverages complex hierarchical convolutional features learned offline from a large dataset.

Note that the proposed method exploits only simple hand crafted feature representation of the target, and achieves competitive performance to the BACF method that utilizes useful background information to train discriminative classifiers. Furthermore, even using only specific target information without learning with auxiliary training data, the proposed method performs well against the deepRedetection and HDT methods, which shows the robustness of the proposed method. The proposed method uses three hand crafted features to provide complementary information for object representation. The compared curves demonstrate the competitive of the proposed method in the DTB70 dataset.

4.4 Low resolution results

4.4.1 LR1 dataset

Figure 8 shows overall performance of the ten trackers in terms of success and precision plots in the LR1 dataset. The proposed method ranks fifth on the success rate of all OPE. There are interesting results that ECO, BACF, SRDCFdecon, fDSST, redetection methods achieve better results than original DTB dataset in both success and precision plots. The proposed method achieves 0.399 and 0.603 in success score and precision score, respectively. The precision score is better than the result (0.586) in the original DTB70 dataset. The other methods degrade slightly in both success and precision scores. The experimental results indicate that some methods can achieve better results in low resolution videos. It should point out that the resolution in LR1 is 640 $\times$ 360, which means the resolution of the target is still large enough.

4.4.2 LR2 dataset

Figure 9 shows overall performance of the ten trackers in terms of success and precision plots. All methods degrade performances in success scores. The deepReDetection drops the smallest value. The SRDCFdecon and BACF drop greatly. In the precision plots, the deepReDetection achieves better results than the results in LR1. All the other methods degrade performance in precision scores. The proposed method achieves 0.388 and 0.583 in success score and precision score, respectively. The resolution in this dataset is 320 $\times$ 180. The experimental results indicate that performances of these trackers are suffered by the low resolution videos.

4.4.3 LR3 dataset

Figure 10 shows overall performance of the ten trackers in terms of success and precision plots. The proposed method ranks fourth on the success rate of all OPE. The proposed method achieves 0.360 and 0.548 in success score and precision score, respectively.

Most of these methods drop larger values than previous situations. The ECO method drops from 0.444 (in LR2) to 0.395. Though the drop is large, it still achieves the first place in success score. The deepReDetection drops from 0.419 (in LR2) to 0.385. The STRCF drops less than the two methods (ECO and deepReDetection). Thus it achieves

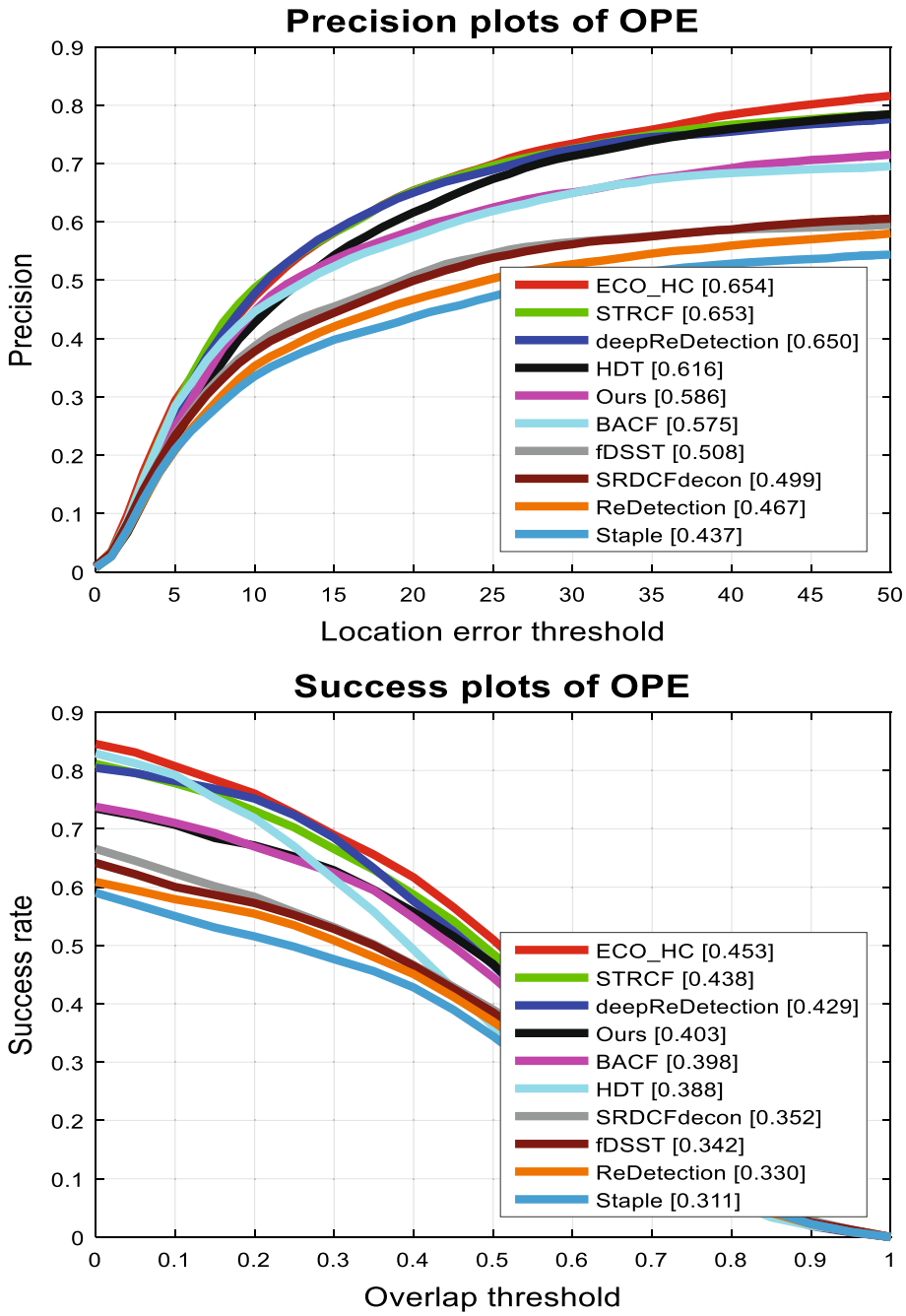

Fig. 7 Precision and success plots for the compared trackers in the DTB70 dataset using OPE

the second place in success score. The HDT method drops from 0.375 (in LR2) to 0.320. This is the biggest drop value in all these methods. SRDCFdecon drops from 0.325 (in LR2) to 0.319. This is the smallest drop value in all these methods.

In the precision plots, the situation is similar to the success plots. The SRDCFdecon improves its results from 0.452 to 0.469. The resolution in this dataset is 160 × 90. The experimental results indicate that performances of these trackers are suffered greatly by the low resolution videos.

4.4.4 LR4 dataset

Figure 11 shows overall performance of the ten trackers in terms of success and precision plots. All these methods degrade performance in this dataset. The ECO method drops from 0.395 (in LR3) to 0.282. It ranks the fourth place in the success score. The deepReDetection drops from 0.385 (in LR3) to 0.296. The STRCF drops smaller than these two methods. Thus it achieves the first place in the success score. The HDT method drops from 0.320 (in LR3) to 0.216. This is the biggest drop value among all

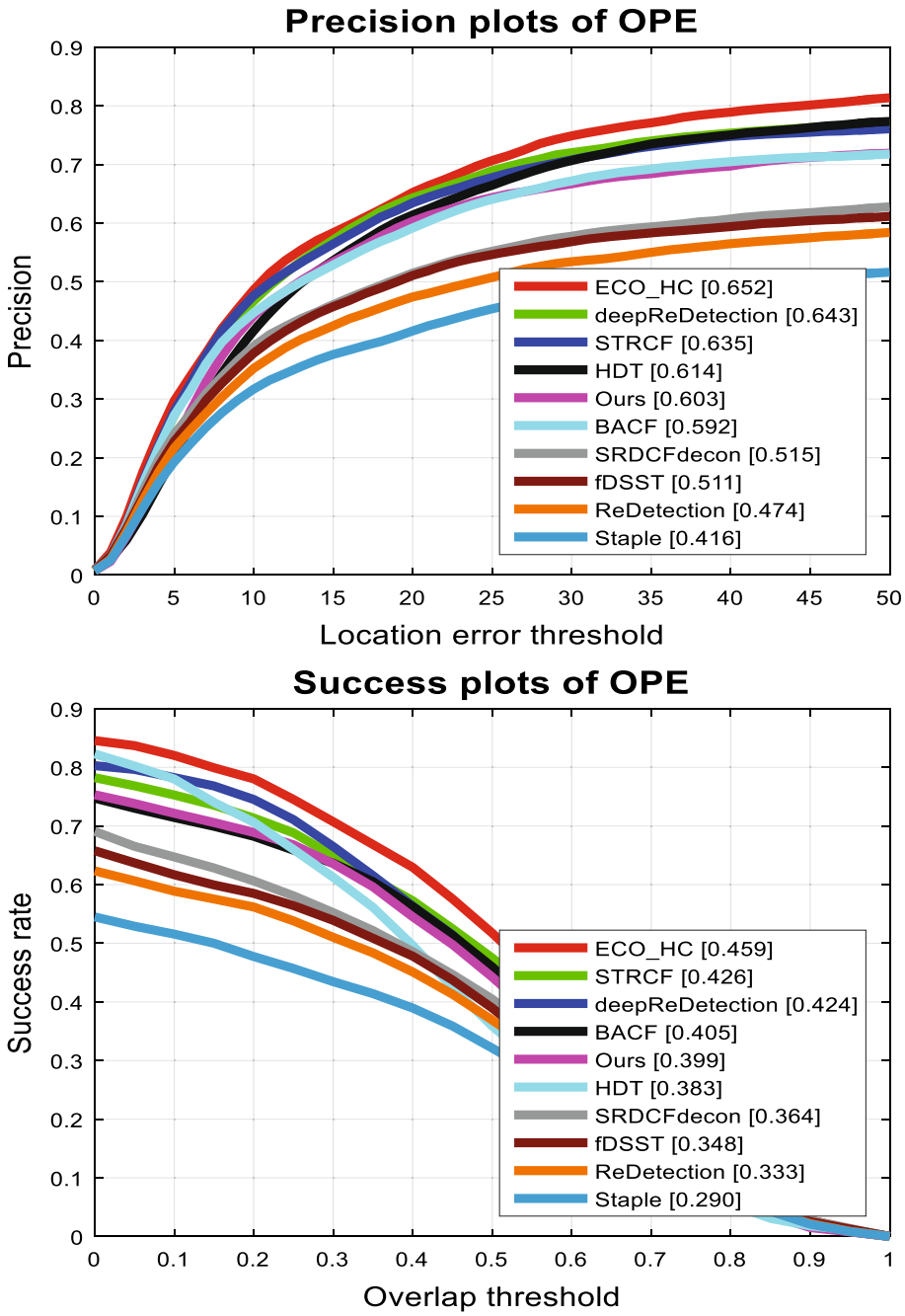

Fig. 8 Precision and success plots for the compared trackers in the LR1 dataset using OPE

these methods. The proposed method achieves 0.309 and 0.484 in success score and precision score, respectively. The success score ranks the second place.

In the precision plots, the situation is similar to the success plots. The resolution in this dataset is 80 × 45. All the targets show ambiguous contours which increase the difficulties of tracking methods. Thus performances of these trackers are suffered greatly by the low resolution videos.

From the original dataset to the LR1 dataset, results of several methods (ECO, BACF, SRDCFdecon, fDSST, redetection methods and the proposed method) are improved. Results of the other methods decrease slightly. This indicates that high resolution videos are not always beneficial to visual tracking.

From the LR1 dataset to LR4 dataset, all the trackers results are reduced. The success score and precision score of the proposed method degrade from 0.403 and 0.586 to 0.309 and 0.484, respectively. This is the smallest reduction in the compared trackers. The success of the proposed method in these datasets is attributed to the following two reasons.

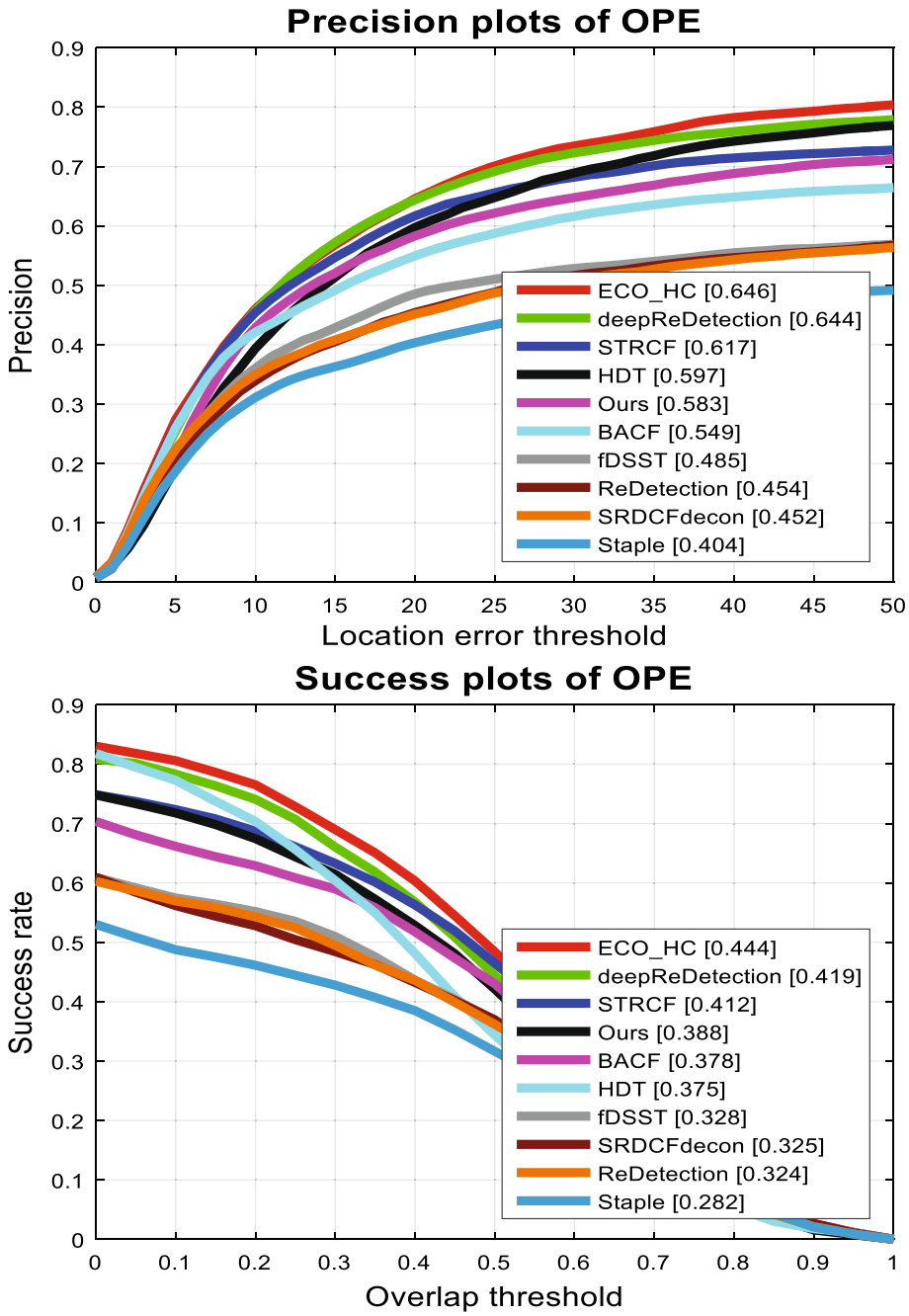

Fig. 9 Precision and success plots for the compared trackers in the LR2 dataset using OPE

Firstly, the multiple feature fusion strategy is effectively to low resolution tracking. The HOG feature and color feature are complementary. Thus, one feature is effective even though another feature fails. In addition, the forward and backward tracking scheme guarantees the proposed tracker to verify tracking quality of each feature.

Secondly, the guided filter is utilized in low resolution tracking. In particular, guided filter is first carried out in low resolution tracking. This procedure enhances image quality which improves object representation.

4.5 Components analysis

In this section, we firstly conduct ablation study to verify the effects of key components. Then the sensitivity of history parameter is tested. Finally, we investigate two types of guided filter.

4.5.1 Ablation study

Three variants of the proposed methods, KCF [28] and DSST [43] are implemented and tested on the low

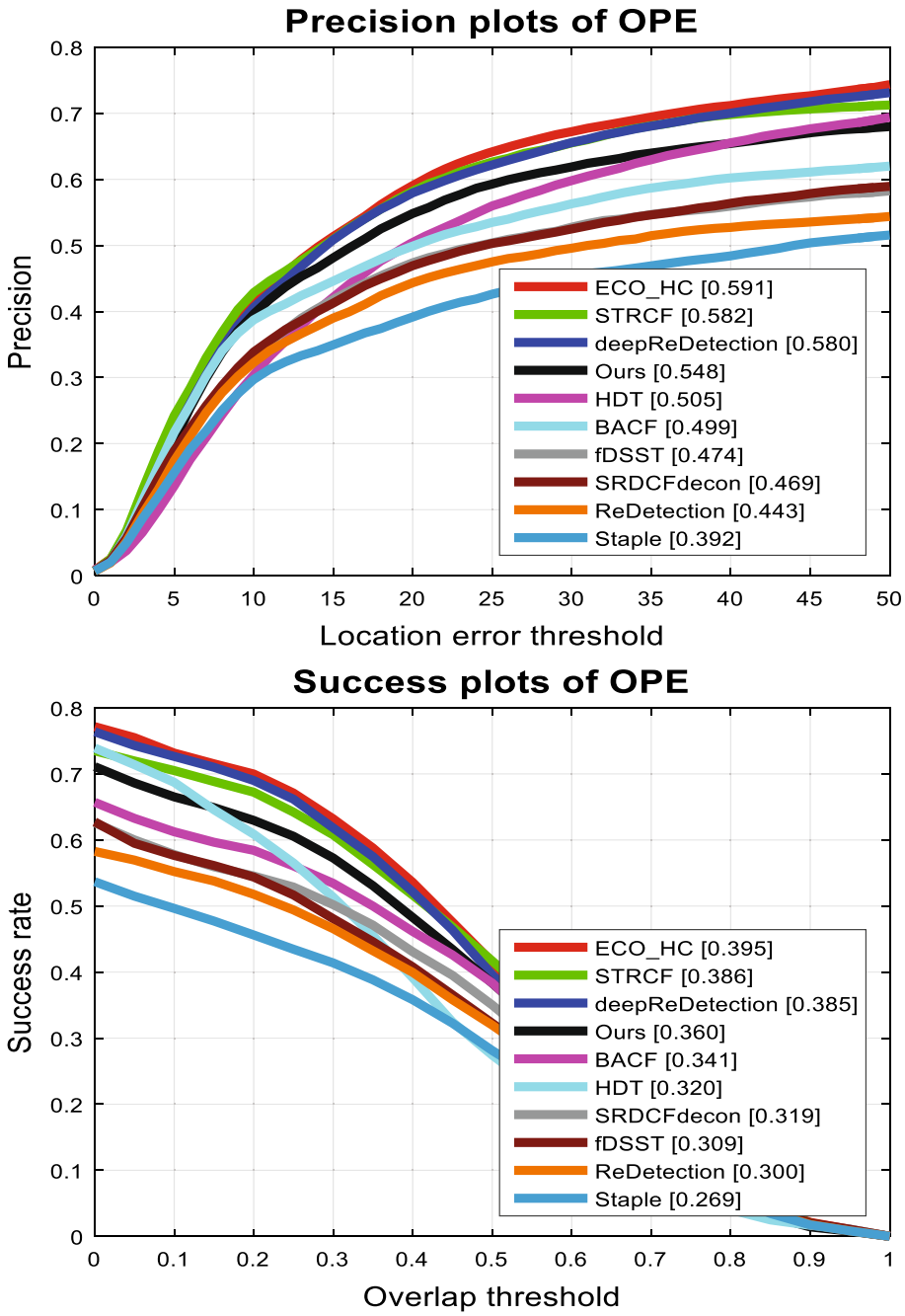

Fig. 10 Precision and success plots for the compared trackers in the LR3 dataset using OPE

resolution datasets to see how the components contribute to improving the final results.

V1: No enhancement in the forward step.

V2: No enhancement in the backward step.

V3: No enhancement in both steps.

V4: baseline KCF.

V5: DSST.

Table 1 shows the success score and precision score for the comparison methods on the low resolution datasets. According to the quantitative comparison, all of the variants of the proposed method perform inferior than the original method. In summary, the comparison results show that all these components are important for constructing a robust tracking algorithm.

In addition, we can observe that the original KCF method and DSST achieve inferior results than the proposed method. The main reasons are as follows. Firstly, our adaptive fusion algorithm effectively integrates HOG and color features to track target accurately and robustly. Secondly, guided filter is employed to enhance the quality of image, which provides accurate appearance description.

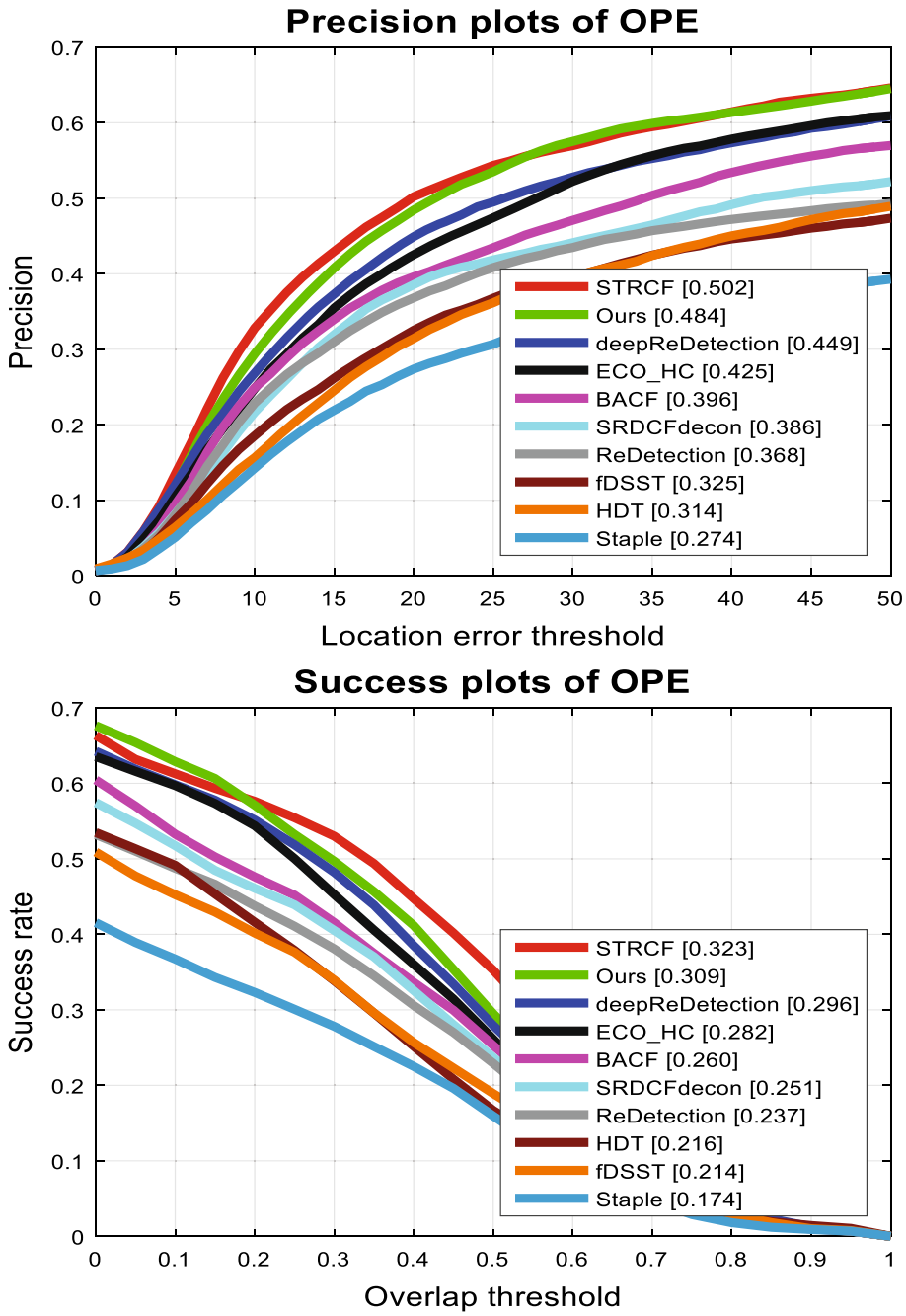

Fig. 11 Precision and success plots for the compared trackers in the LR4 dataset using OPE

Table 1 The success score and precision score for the comparison methods on the low resolution datasets

	LR1	LR2	LR3	LR4
Ours	0.399, 0.603	0.388, 0.583	0.360, 0.548	0.309, 0.484
V1	0.373, 0.542	0.378, 0.544	0.352, 0.519	0.270, 0.410
V2	0.392, 0.592	0.378, 0.565	0.360, 0.529	0.307, 0.484
V3	0.395, 0.583	0.385, 0.565	0.358, 0.534	0.292, 0.460
V4	0.282, 0.473	0.271, 0.447	0.253, 0.418	0.196, 0.318
V5	0.323, 0.464	0.325, 0.479	0.269, 0.393	0.206, 0.296

Thirdly, forward and backward tracking scheme further guarantees tracking accuracy.

Table 2 The success score and precision score for the history parameter on the low resolution datasets

	LR1	LR2	LR3	LR4
Ours	0.399, 0.603	0.388, 0.583	0.360, 0.548	0.309, 0.484
V6	0.393, 0.585	0.380, 0.567	0.355, 0.542	0.274, 0.430
V7	0.387, 0.586	0.384, 0.576	0.345, 0.521	0.294, 0.464

4.5.2 Parameter analysis

In this section, we investigate the sensitivity of parameter (history parameter) in our method. For fair comparison, all the other components and parameters are fixed when evaluating history parameter.

V6: history = 3.

V7: history = 1.

Table 2 demonstrates the success score and precision score for the history parameter on the low resolution datasets. We observe that the results change slightly with different history parameters according to the results in Table 2. The comparison results demonstrate the robustness and effectiveness of the proposed method.

4.5.3 Guided filter analysis

In this paper, we adopt the guided filter in [16]. We also implement the guided filter in [15] and [17]. For fair comparison, all the other components and parameters are fixed. Table 3 shows the success score and precision score for different types of guided filter on the low resolution datasets. We can observe that the two methods [15, 17] all achieve appealing results.

V8: guided filter in [15].

V9: guided filter in [17].

The guided filter in [16] runs faster than the method in [15]. And the method in [17] runs a little slowly. Thus, the method in [16] is chosen in the tracking framework.

4.6 Quality evaluation

Figure 12 shows some sampled results in the sequence of Animal1 in which the target objects undergo large illumination variations. The first row is the original sequence, the second row, the third row, the fourth row and the fifth row are the LR1, LR2, LR3 and LR4, respectively. In the original sequence, the fDSST, HDT and Staple methods track around the target. The other methods drift away from the target when scale variation and illumination change occurs. In the LR4 sequence, the ECO, HDT and SRDCFdecon methods drift away to the background.

Table 3 The success score and precision score for different types of guided filter on the low resolution datasets

	LR1	LR2	LR3	LR4
Ours	0.399, 0.603	0.388, 0.583	0.360, 0.548	0.309, 0.484
V8	0.376, 0.566	0.381, 0.565	0.354, 0.531	0.292, 0.456
V9	0.399, 0.594	0.381, 0.569	0.345, 0.518	0.282, 0.451

Figure 13 shows sampled results of the sequence Horse1 where the targets undergo heavy occlusions. In the original sequence, a horse is almost fully occluded by a tree. Only the ECO and SRDCFdecon algorithms are able to track the object when the horse reappears in the screen. In the LR4 sequence, only our method persistently tracks the object from the beginning to the end. The other methods do not perform well.

4.7 Discussions

There is not much change for the overall performance of most tracking methods in the LR1 dataset compare to the performance of original DTB70. This demonstrates that these tracking methods are somewhat stable on videos in the LR1 dataset.

In the LR2 dataset, most of these trackers are suffered from low resolution videos. In the LR3 and LR4 datasets, the performances of these trackers degrade greatly. The STRCF achieve better results in the LR4 dataset than the ECO method, while the ECO method achieves better results than the other methods in the rest datasets. The deepReDetection and HDT methods drop big values in the LR4 shows that the generic features trained offline from numerous auxiliary datasets may not adapt well to object appearance variations in such low resolution tracking.

The primary goal of this paper is to gain a deeper understanding into the UAV based low resolution tracking. Thus, inevitably, some excellent algorithms are not represented in the comparison and not integrated into our tracking system.

The tracking performance can be attributed to four factors. Firstly, the enhanced image alleviates the effect of low resolution videos. The image quality is a key issue to object representation. Secondly, the proposed algorithm exploits a multiple features scheme to improve the object representation. These features are complementary to each other to cope with deformation, background clutter and so on. Thirdly, the adaptive fusion algorithm considers all tracking positions of different features, which improves robustness of the tracking results. Fourthly, the forward and backward tracking scheme is integrated into correlation filter tracking framework to avoid model drift.

Although our tracker demonstrates appealing performance in the low resolution tracking datasets, it is still far from perfect. To address this challenging problem, we think there are two possible solutions.

First, low resolution dataset can be further studied. A great number of trackers are developed while low resolution situation is one of the greatest challenges and a largely under explored domain. Observed from the results in the

Fig. 12 Screenshots sampled results from five different resolutions in Animal1 sequences

original dataset and the LR1 dataset, it motivates us to find a suitable resolution for visual tracking.

Second, deep learning based methods are reported to be implemented in mobile devices. Deep features which are more representative than hand-crafted features are another choice to improve low resolution tracking performance.

However, these solutions are beyond the scope of this paper, we would like to exploit them in our future work.

5 Conclusion

In this paper, we have addressed the problem of low resolution tracking in UAV situations. The objective of this study is to investigate the performance of different trackers on four low resolution datasets. We have arrived at some interesting conclusions. First, recently proposed tracking methods are studied in this article. These trackers suffered differently in the four low resolution datasets. Second, HOG and color names features in encoding low resolution

Fig. 13 Screenshots sampled results from five different resolutions in Horse1 sequences

images are effective. And the fusion method can affect the result significantly. Third, image enhancement technique is employed to improve the performance. Our work enlightens several interesting directions to pursue, including the usage of image processing technique in visual tracking and advanced fusion method. It is our hope that, besides the general visual tracking which has been the focus of many studies, low resolution and other equally important tracking situations will attract more research attention as a consequence of our work.

Acknowledgement This work was partially supported by National Science Found for Young Scholars under Grant No. 61806186, State Key Laboratory of Robotics and System (HIT) under Grant No. SKLRS-2019-KF-15, the program 'Construction of Fujian Research Institute on Intelligent Logistics Industry Technology' under Grant No. 2018H2001, CAS Pioneer Hundred Talents Program (Type C) under Grant No. 2017-122, and the program 'Quanzhou Science and Technology Plan' under Grant No. 2019C112, No. 2019C011R and No. 2019STS08.

Compliance with ethical standards

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work. There is no professional or other personal interest of any nature or kind in any product, service or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for UAV tracking. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision – ECCV 2016, vol 9905. Lecture notes in computer science. Springer, Cham
2. Siyi L, Yeung D-Y (2017) Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models. AAAI 122:4140–4146
3. Dawei D, Qi Y, Yu H, Yang Y, Duan K, Li G, Zhang W, Huang Q, Tian Q (2018) The unmanned aerial vehicle benchmark: object detection and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 370–386
4. Pengfei Z, Wen L, Bian X, Ling H, Hu Q (2018) Vision meets drones: a challenge, pp 1–11. arXiv Prepr, arXiv:1804.07437
5. Lu D, Yong W, Robert L, Xin-Bin L, Fu S (2018) Scale-aware RPN for vehicle detection. ISVC 12:487–499
6. Jianan L, Xiaodan L, ShengMei S, Xu T, Jiashi F, Yan S (2017) Scale-aware fast R-CNN for pedestrian detection. IEEE Trans Multimed 20(4):985–996
7. Jiang N, Heng S, Liu W, Ying W (2012) Discriminative metric preservation for tracking low-resolution targets. IEEE Trans Image Process 21(3):1284–1297
8. Zhiguan L, Yuan C (2018) Robust visual tracking in low-resolution sequence. In: 25th IEEE International Conference on image processing (ICIP), IEEE, pp 4103–4107
9. Martin D, Bhat G, Khan FS, Felsberg M (2017) ECO: efficient convolution operators for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6638–6646
10. Fang Y, Yuan Y, Li L, Jinjian W, Lin W, Li Z (2017) Performance evaluation of visual tracking algorithms on video sequences with quality degradation. IEEE Access 5:2430–2441
11. Navneet D, Triggs B (2005) Histograms of oriented gradients for human detection. In: International Conference on computer vision pattern recognition (CVPR'05), IEEE Computer Society, vol 1, pp 886–893
12. Ning W, Zhou W, Tian Q, Hong R, Wang M, Li H (2018) Multi-cue correlation filters for robust visual tracking. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4844–4853
13. Ma C, Huang J-B, Yang X, Yang M-H (2015) Hierarchical convolutional features for visual tracking. Proc IEEE Int Conf Comput Vis 2102:3074–3082
14. Kristan M, Matas J, Leonardis A, Felsberg M, Cehovin L, Fernandez G, Vojr T, Hager G, Nebehay G, Pflugfelder R (2015) The visual object tracking VOT2015 challenge results. Proc IEEE Int Conf Comput Vis Workshop 2015:564–586
15. He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35(6):1397–1409
16. Lu Z, Long B, Li K, Fajin L (2018) Effective guided image filtering for contrast enhancement. IEEE Signal Process Lett 25(10):1585–1589
17. Guo X, Li Y, Ma J, Ling H (2020) Mutually guided image filtering. IEEE Trans Pattern Anal Mach Intell 42(3):694–707
18. Smeulders AWM, Chu DM, Cucchiara R, Calderara S, Dehghan A, Shah M (2014) Visual tracking: an experimental survey. TPAMI 36(7):1442–1468
19. Wu Y, Lim J, Yang M-H (2015) Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell 37(9):1834–1848
20. Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R, Cehovin L, Vojr T, Hager G, Lukezic A, Fernandez G (2016) The visual object tracking VOT2016 challenge results. Proc Eur Conf Comput Vis Workshop 9914:777–823
21. Liang P, Blasch E, Ling H (2015) Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans Image Process 24(12):5630–5644
22. Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Bai H, Xu Y, Liao C, Ling H (2019) LaSOT: a high-quality benchmark for large-scale single object tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)
23. Yong W, Robert L, Daniel L, Ors AO, Xu X, Zhu C (2018) Deep convolutional correlation filters for forward-backward visual tracking. ISVC 132:320–331
24. Qi Y, Zhang S, Qin L, Yao H, Huang Q, Lim J, Yang M-H (2016) Hedged deep tracking. Proc IEEE Conf Comput Vis Pattern Recogn 9810:4303–4311
25. Feng L, Tian C, Zuo W, Zhang L, Yang M-H (2018) Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4904–4913
26. Sun Z, Wang Y, Laganiere R (2019) Hard negative mining for correlation filters in visual tracking. Mach Vis Appl 30(3):487–506
27. Bolme DS, Beveridge JR, Draper B, Lui YM (2010) Visual object tracking using adaptive correlation filters. Proc IEEE Conf Comput Vis Pattern Recogn 9123:2544–2550
28. Henriques J, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
29. Danelljan M, Häger G, Khan FS, Felsberg M (2017) Discriminative scale space tracking. IEEE Trans Pattern Anal Machine Intell 39(8):1561–1575
30. Luca B, Valmadre J, Golodetz S, Miksik O, Torr HSP (2016) Staple: complementary learners for real-time tracking. CVPR 8943:1401–1409
31. Danelljan M, Hager G, Khan FS, Felsberg M (2016) Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. Proc IEEE Conf Comput Vis Pattern Recogn 8930:1430–1438
32. Danelljan M, Hager G, Khan FS, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. Proc IEEE Int Conf Comput Vis 9420:4310–4318
33. Collins R, Zhou X, Teh SK (2005) An open source tracking testbed and evaluation web site. In: IEEE Int workshop on performance evaluation of tracking and surveillance, pp 17–24
34. Galoogahi K, Fagg HA, Lucey S (2017) Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE Conference on ICCV, pp 1135–1143

35. Ning W, Wengang Z, Li H (2018) Reliable re-detection for long-term tracking. In: IEEE transactions on circuits and systems for video technology
36. Danelljan M, Khan FS, Felsberg M, van de Weijer J (2014) Adaptive color attributes for real-time visual tracking. Proc IEEE Conf Comput Vis Pattern Recogn 9872:1090–1097
37. Mei X, Ling H (2011) Robust visual tracking and vehicle classification via sparse representation. IEEE Trans Pattern Anal Mach Intell 33(11):2259–2272
38. Wei X, Shen H, Kleinsteuber M (2019) Trace quotient with sparsity priors for learning low dimensional image representations. In: IEEE transactions on pattern analysis and machine intelligence, pp 1–17
39. Wang Y, Shiqiang H, Shandong W (2015) Visual tracking based on group sparsity learning. Mach Vis Appl 26(1):127–139
40. Kim H-U, Lee D-Y, Sim J-Y, Kim C-S (2015) Sowp: spatially ordered and weighted patch descriptor for visual tracking. In: Proceedings of IEEE International Conference on computer vision
41. Li C, Lin L, Zuo W et al (2018) Visual tracking via dynamic graph learning. IEEE Trans Pattern Anal Mach Intell 41(11):2770–2782
42. Li C, Liang X, Lu Y, Zhao N, Tang J (2019) RGB-T object tracking: benchmark and baseline. Pattern Recognit 96:106977
43. Danelljan M, Hager G, Khan F, Felsberg M (2014) Accurate scale estimation for robust visual tracking. In: British machine vision conference, Nottingham, September 1–5, BMVA Press