Multimedia Tools and Applications (2021) 80:16367–16395
https://doi.org/10.1007/s11042-020-08791-z

Multi-features guided robust visual tracking

Yun Liang[1] · Jian Zhang[2] · Mei-hua Wang[1] · Chen Lin[3] · Jun Xiao[4]

Received: 5 May 2019 / Revised: 17 January 2020 / Accepted: 24 February 2020 /
Published online: 23 May 2020

Abstract
This paper focuses on dealing with the tracking challenges such as target occlusion and deformation. It proposes a new tracking method via extracting and evaluating multi-features for both target region and its adjacent surroundings. The multi-features separately describe the key factors to detect target including the color feature, the shape and contour feature, and the distributions of structure and intensity described by the Pearson Correlation Coefficient. These multi-features are proposed as the basic representation of target template and candidates and used to define a matching algorithm between them. The best matched candidate is taken as the final tracking result. To improve the efficiency of target template and candidates, the region of importance (ROI) for target is proposed by evaluating the distribution of salient values on many extended regions. The ROIs produce more accurate regions to form target template and candidates. Finally, a new template update method is defined based on the precision of tracked result to adapt to target state and achieve the follow target tracking. Using 25 videos in visual tracking benchmark, we achieve the quantitative and qualitatively evaluations of 12 different trackers. Many experiments demonstrate that our tracker produces much better results than the present trackers in dealing with target occlusion, deformation, rotation, background clutters.

Keywords Visual tracking · Multi-features · Target template · Template matching method · Update template

CLC number: TP 37

✉ Jian Zhang
jeyzhang@outlook.com

[1] College of Mathematics and Informatics, South China Agriculture University, Guangzhou 510642, China

[2] School of Science and Technology, Zhejiang International Studies University, Hangzhou 310023, China

[3] Department of Computer Science, Xiamen University, Xiamen 361005, China

[4] College of Computer Science & Technology, Zhejiang University, Hangzhou 310027, China

1 Introduction

Visual tracking is a hot topic in computer vision [28]. It can compute the location and size of a moving target in a video. Recently, visual tracking is widely used in surveillance, automatic drive, drone, virtual reality and so on. However, the disturbs from complex backgrounds and drastic target deformations bring many challenges in tracking such as illumination change, object deformation, occlusion and so on. Usually, many challenges emerge simultaneously, and this leads the robust and fast visual tracking still under successfully solved [36].

Recently, many trackers have been proposed [36]. Some trackers are defined based on learning methods [5, 30], and some others are based on other schemes such as the correlation filters [2, 11, 18, 33] and sparse coding [39]. The trackers via learning usually are limited by the training model, the accurate parameters and the training data. The trackers without learning need suitable and smart scheme to represent and forecast target. Many researches have demonstrated the features from multi-fields of target perform effectively in identifying target as the multi-features provide different key cues of target by describing target from different aspects [31]. Therefore, we define multi-features to construct target template, and utilize them on both ROI about target and the target template to detect the accurate region of target object.

This paper proposes to utilize multi-features from both the target template and its surroundings to detect target object. The multi-features are defined to evaluate how similar to target template that a candidate is. Our multi-features include the *color feature* describing the color distribution, the *shape feature* describing shape and contour of target object, and the *Pearson correlation coefficient featur*e about the structure and intensity distribution of target. Meanwhile, we propose the ROI about target by heuristic probing all the probable target regions, which greatly improves the effectiveness of candidates by rejecting redundant ones. By our multi-features, we track a target via its important cues such as color, shape, contour, structure and intensity to get the accurate result. Compared with 11 present trackers, our tracker produces more favorable results by the quantitative and qualitative evaluations on the tracking Benchmark 2013 [36].

2 Related work

Visual tracking plays an important role in computer vision for its good property in computing the continuously changing states of moving object. The unpredictable and complex variations of target and its surroundings bring many kinds of tracking challenges including background clutter, deformation, occlusion and so on. Although many trackers have been proposed recently, they still cannot successfully deal with the challenges especially when they are emerging at the same time. Most present trackers depend on some specified kind of target feature to search and identify target object. According to the corresponding scope of target features, the present trackers can be classified into two kinds, including the ones based on local features [38] and the other ones based on global features [12, 13, 35].

For the trackers based on local features, they described the target by features extracted from the local regions or components of target. For example, Adam et al. [1] use the features of irregular patches from target region to represent target, and the target state is computed via the vote of these patches. Jia et al. [14] suggest dividing the target into regular patches and use sparse coding to achieve tracking. Vojir et al. [34] select the features from the successful tracked results to detect target. Zhang et al. [38] utilize the features by HOG to describe image

patches and used SVM to predict target. Kwon et al. [17] propose to deal with target deformation by updating the representation and topology of its local patches. Cai et al. [4] employ a set of superpixels to describe target and define the match algorithm of dynamic graph to achieve tracking. Recently, more and more people try to capture the information of video and image by learning method. For example, Liu et al. employed the multi-level learning method [21] to analyze the image caption [20] and the video caption [37]. In tracking field, Guo et al. [9] use the trained local features to construct and update the appearance representation of target to deal with occlusion and deformation. Grabner et al. [8] track target by a classifier to match the key points about target between different frames. Godec et al. [7] train an on-line Hough forest to compute the local regions of target. Maresca et al. [24] utilize the Hough variation and key points to track target. Many trackers based on local features have demonstrated efficiently in dealing with some challenges such as partial occlusion. However, this kind of trackers is not robust especially in dealing with great deformation and occlusion. The main reason is that local features often have great and fast variations to adapt to target appearance change. These variations introduce many errors which finally lead to tracking failure or drift.

For the trackers of global features, they get features from the whole target region or a candidate and detect target by evaluating these global features. For example, Zhu et al. [41] propose to combine gradient and color histogram to form the global feature to identify target. Zhu et al. [42] define the global feature via the contours and the structure information of target. Hong et al. [13] construct the global feature by the information of target recorded at three-time intervals, namely the instant memory, the short-time memory and the long-time memory. To match the global features between target and candidates, different people suggest different methods. For example, Bolme et al. [3] suggested to use the least square error and the correlative filters to adapt to the changes of appearance such as the scale variation of target. Henriques et al. [11] advise to utilize the Least square classifier and the kernel related filter (KCF) to achieve tracking. They improved the tracking accuracy and reduced the implement time by using the cyclic matrix to describe the dense samples around the predicted target region. The trackers based on KCF get target very quickly at even support the on-line tracking process, but they sometimes introduced tracking drift especially when dealing with complex surroundings or occlusion of target. Some people improved KCF by defining multi-features of target [23] or a new fusing scheme [22]. However, many experiments have demonstrated that the trackers based on global features perform favorable in describing the whole changes of target especially when they use more than one kind of target feature. However, this kind of trackers is very sensitive to the predicted region about target which is used to produce samples [36]. When the predicted region covers too much background, the results will drift from the ideal target region, but when it covers only part of target, the results may become totally failed.

Therefore, this paper proposes a new tracker by predicting an accurate searching region of target and constructing multi-features to describe target. There are two advantages of our tracker. First, we define an algorithm of ROI based on salient values to predict the region which covering target and being robust for candidates. This ROI improves the tracking efficiency and reduces the implement time by accurately predicting the searching region of target. Second, we define multi-features from different aspects of target object to reduce the disturbs from background and improve tracking accuracy. Compared with the present trackers, especially with the Deep Learning (DL), our method successfully improves the tracking accuracy without high time cost and some pre-training which usually happens on the recent DL trackers.

3 The proposed tracker

This section describes the framework of our tracker as shown in Fig. 1 and the details how to construct it.

3.1 The framework of our tracker

This paper proposes a new tracker by matching the multi-features between target template with samples and employing the ROI of target to design our effective candidates. Our tracker is achieved by the following five steps one by one as shown in Fig. 1. First, we compute the ROI of target (as Fig. 1(c)) by evaluating the salient values of many extended regions of target (as Fig. 1(b)) and use the ROIs to get the optimal extended region (OER). Second, we construct a target template by the multi-features of the first specified target region and its ROI as shown in Fig. 1(d). Third, we transmit the current optimal extended region to the next frame and use its ROI to optimize the candidates. The red points in the bottom of Fig. 1(e) are the selected optimal candidates' centers from the points in the top. Fourth, we match target template with candidates (as Fig. 1(f)) to get tracked result (as Fig. 1(g)). Finally, we update the target template (as Fig. 1(d)) based on new result to adapt to the changes of target and its surroundings to do following tracking. The main processes of our tracker:

(1) Compute the ROI of target on the first frame. By the specified target on the first frame, we construct many extended regions by adding some background around the target region. Then, we take the connected region with the biggest salience and big area of each extended region as its ROI such as the white regions in the second column of Fig. 1(b). The ROI most similar to the target is defined as the ROI of target, and the extended region with it is the Optimal Extended Region in which the target is preserved as the most salient object.
(2) Construct the target template. The ROI of target describes the region of our target template. We use its multi-features (including color feature, shape feature, Pearson correlation coefficient feature about the structure and intensity) to describe target template.
(3) Get efficient candidates. By the OER of last frame, we first obtain the OER of current frame. Then, we sample candidates around its ROI and discard the invalid samples to get efficient candidates.

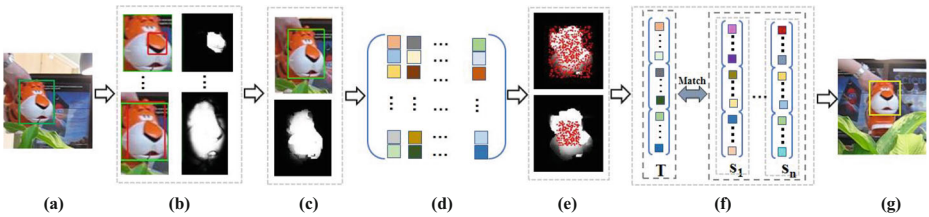

Fig. 1 The Flowchart of our method. First, we input the first frame with the initialized target region (a) and compute many ROIs (b) to search the ROI of target (c). Second, we construct target template by computing the multi-features of target as (d) and implement the sampling (f) based on (c). Third, we evaluate samples by matching the multi-features between them (the $s_1,\ldots,s_n$ in (f)) and the target (T in (f)). Finally, we get the tracking result (g) by the above evaluation

(4) Match target template with candidates. We evaluate the differences between the multi-features of target template and a candidate to achieve the matching. The matched candidate with the minimum distance is taken as the tracking result.
(5) Update target template. To adapt to the changes of target and its surroundings, we update target template by updating its multi-features with the multi-features from new tracked result. An update factor is defined to decide whether and how much we need update.

3.2 Construct the target template

The target template region is first specified on the first frame. For the following frame, it is computed by four steps. First, we compute the extended regions of target. Second, for each extended region, we calculate its ROI. Third, by comparing the features about shape and center locations of target from the ROIs and target region, we get the OER about target. Finally, we construct the target template by extracting its multi-features to describe target.

(1) Compute the extended regions

With the target region on the first frame, we compute the extended regions of target by expanding the target region such as the dotted rectangles in Fig. 2(b). The expansions are achieved by adding some surrounding background along different directions. An expansion is done by (Lexp, Rexp, Texp, Bexp). The Lexp describes the displacement between the left boundaries of an extended region and target region while the Rexp is the displacement of the right boundaries, the Texp is the displacement of the top boundaries and the Bexp is the displacement of the bottom boundaries. In our experiments, Lexp is max (0,Cx-Width*Xratio), Rexp is min (Cx + Width*Xratio,Xsize). Texp is set to min(Cy + Height*Yratio,Ysize). Bexp is computed by max(0, Cy-Height*Yratio). Here, (Cx, Cy) are the center of target region, and (Width, Height) are its width and height. The (Xsize, Ysize) are the boundary value of the video frame. The (Xratio, Yratio) decides the ratio to enlarge the target region. In our experiments, we set the ratios to be (1.2,1.2) to control the ROI not too big or small (Fig. 3).

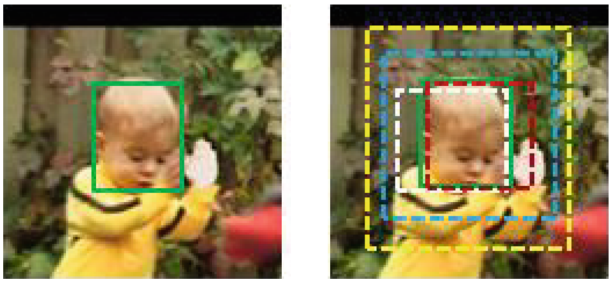

Fig. 2 The extended regions of target. (a) green rectangle denotes target, (b) dotted rectangles are extended regions

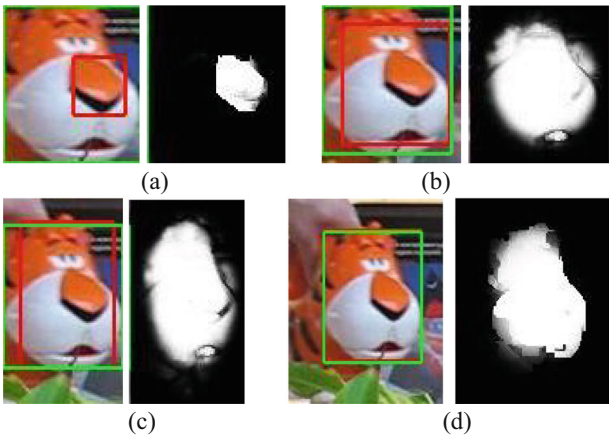

Fig. 3 The ROIs of extended regions. The left images in (a-d) are the extended regions while the right ones are their ROIs. The red rectangle describes the ROI while the green rectangle describes the target region

(2) Calculate the ROIs of extended regions

For an extended region, we use the method proposed by Tu [32] to get the salient values of its pixels. This saliency is defined based on the minimum spanning tree and the geodesic distance. It produces more accurate and faster saliency than the traditional saliency methods. With this method, bigger saliency means more possible to be object. We select the connected region with the biggest saliency and biggest area of an extended region as its region of importance (ROI). Here, we do a binarization process on the saliency map to delete the pixels with small salient value. Following this, the ROI has the maximum probability to be the target. We use the 4-connected method to justify the connected region.

(3) Get the optimal extended region about target

This paper defines a similarity measure to compute the matched degree between the ROIs and the specified target region. The extended region whose ROI is most similar to target is taken as the optimal extended region. The similarity measure is combined by:

- The differences of the width and height between target rectangle (as green rectangles in Fig. 4(a-d)) and the bounding box of a ROI (as red rectangles in Fig. 4(a-d)).
- The differences between the centers of target rectangle and the bounding box of a ROI.
- The combination of the above differences with two coefficients to adjust their weights.

It is defined as:

$$D_{sim} = \alpha_1 \times S + \alpha_2 \times E \tag{1}$$

where S defined in Eq. 2 is the shape distance to describe the first difference and E defined in Eq. 3 is the center distance to describe the second difference. α_1, α_2 are weights. According to our many experiments, we get that ($\alpha_1 = 0.2, \alpha_2 = 0.8$) leads to good performance. Therefore, we set the two factors based on the experiment values.

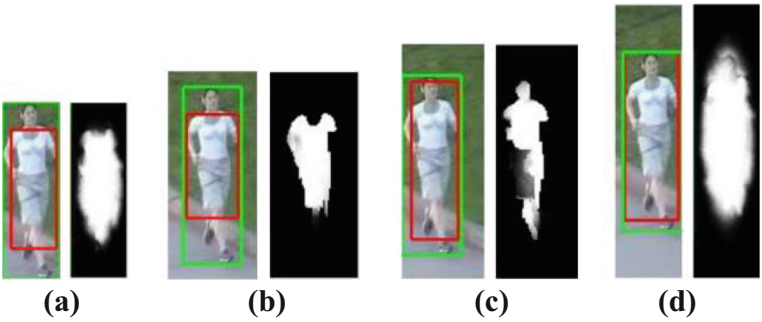

Fig. 4 The ROIs of extended regions. The left images in (a-d) are the extended regions while the right ones are their ROIs. The red rectangle describes the ROI while the green rectangle describes the target region

If (N_h, N_w) are the height and width of the target rectangle and (M_h, M_w) are the height width of the bounding box of a ROI. The shape distance S is:

$$S = |M_h - N_h| + |M_w - N_w| \tag{2}$$

If $(P(n,x), P(n,y))$ describe the x-coordinate and y-coordinate of the center of target rectangle and $(P(m,x), P(m,y))$ are denoted as the x-coordinate and y-coordinate of the center of the bounding box of a ROI. The center distance E by:

$$E = \sqrt{(P(m,x) - P(n,x))^2 + (P(m,y) - P(n,y))^2} \tag{3}$$

(4) Construct target template via multi-features

We construct the target template by the multi-features of the target region. Three kinds of features are utilized to describe the target template. One is the color feature which is defined by the histogram of the HSV color information on the target region, which reflects the main distribution about color information of target. Another one is the shape feature which describes the object area with irregular contour of the ROI in the optimal extended region of target such as the white region in Fig. 4(d). The third one is the Pearson correlation coefficient between the target and a candidate which describes the difference about the structure and intensity distribution between target and candidate. These multi-features describe the target from different aspects and provide robust descriptions of target.

3.3 Get candidates via transmitting and filtering

To sample efficient candidates, we first compute the ROI on the current frame by transmitting the optimal extended region of the last frame. Then, we densely sample candidates by randomly selecting their center points in the ROI. Finally, we filter out some invalid candidates to improve accuracy and reduce time cost.

First, we compute the ROI on the current frame According to the optimal extended region in section 3.2, the target on the current frame locate in its ROI. However, the OER is calculated based on the known target region on the same frame. Therefore, we cannot get it on current frame because the target region is unknown. For the OER changes very little in adjacent frames, we transmit it from the last frame to the current frame. It means the OER on the current

frame has the same center position, width and height with the last one. Then, we use the method in [32] to compute the ROI of target on the current frame by this OER.

Second, we construct candidates based on the ROI With the bounding box of ROI on current frame, we randomly sample dense points in it as the center positions of candidates. The width and height of a candidate are the same with the last target. When the boundary of a candidate across the boundary of the current optimal extended region, we denote it as an invalid sample. This kind of candidates often covers too much background with too little object, and drifts away from the ROI. By filtering out all the invalid candidates from the initial samples, we form the final candidates to detect target (Fig. 5).

3.4 Evaluating candidates by their multi-features

This paper defines a match measure to compute the similarity between a candidate and target template. The candidate best matched to target template is the tracking result on current frame. Our match measure D is defined by matching the multi-features about target and its candidates, including: the items of color feature match D_c, shape feature match D_s, structure and intensity match D_r computed by the Pearson correlation coefficient. We define D by:

$$D = \beta_1 \times D_c + \beta_2 \times D_s + \beta_3 \times D_r \tag{4}$$

where β_1, β_2,β_3 are the coefficients and satisfy ($\beta_1 + \beta_2 + \beta_3 = 1$). In this paper, we set $\beta_1 = 0.25$, $\beta_2 = 0.25$,$\beta_3 = 0.5$ based on experiments. Bigger value of D means better matched. The best matched sample is the sample owns the biggest value of D. Compared with the other image feature such the Haar, LBP and HOG, our multi-features can describe target object more accurate from both local texture details by color feature, and the global object level measure by the shape feature and person correlation coefficient. The advantages using both local feature and global features has been demonstrated in many work [40]. The D_c, D_s and D_r are defined as follows.

- The color feature match D_c

This paper utilizes the histogram of color information in HSV space to describe the color feature of target template and the candidates. By this histogram, our color feature describes the

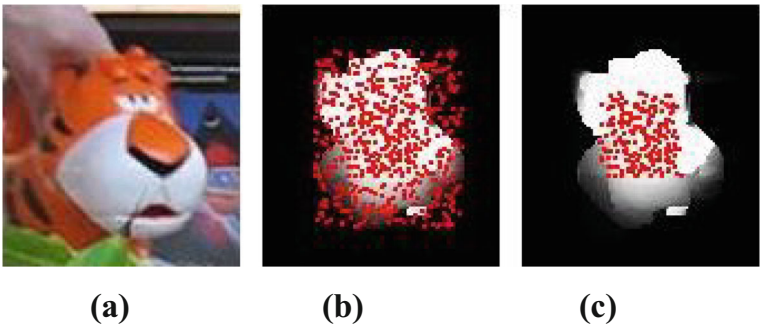

Fig. 5 Sampling the candidates. (a) is the optimal extended region of target, the red points in (b) and (c) are the centers of candidates before and after filtering

ratios of different colors and is irrelevant with space information. It is a set of the histograms of different color ranges, and the histogram of color range i is:

$$H(i) = n_i / \sum_{i=0}^{k-1} n_i \quad (i = 0, 1, 2, ..., k-1) \tag{5}$$

Where i is the color range describing the $i'th$ interval of color values. The n_i is the number of pixels in range i. The k is the number of color ranges.

If we use A_i andB_i to separately describe the histogram of color range i for a candidate and the target. We define the color feature match D_c by:

$$D_c = 1 - \frac{\sum_{i=1}^{k} A_i B_i}{\sqrt{\sum_{i=1}^{k} (A_i)^2} \sqrt{\sum_{i=1}^{k} (B_i)^2}} \tag{6}$$

- The shape feature match D_s

According to the calculation about the ROI of target, we known the ROI is the region only covering the target object without any background. Therefore, if a candidate covers larger area of ROI, it includes both target and background. As following the above sampling method, each candidate includes part of the ROI. We define the part of the ROI included in a candidate is its important region. Then we define the shape feature match by comparing the important region of a candidate to the ROI of target as:

$$D_S = \frac{S_{tar} \cap S_{can}}{S_{tar} \cup S_{can}} \tag{7}$$

Where S_{tar} and S_{can} are the areas of ROI of target and the important region of a candidate. In this paper, we compute the area by counting its pixels, namely the pixel values included in each area is set to its area value. $\cap$ is the intersection of the two areas and $\cup$ is the union of them. $S_{tar}$is computed by the target region on last frame. When D_S is small, it means the candidate includes much background and little target. Otherwise, it means the candidate includes little background and much target, which is more possible to be the target region for the current frame.

- The structure and intensity match D_r

We define the structure and intensity match by comparing the matrix between the target and a candidate using the Pearson correlation coefficient. As the matrix values describe the changes of color value pixel by pixel, these changes carefully describe the structure and intensity of the target and candidates. If the matrix of a candidate is more matched with that of target, the candidate has bigger probability to be target. Therefore, we define the structure and intensity match by the Pearson correlation coefficient which demonstrates effectiveness in comparing two matrixes [27].

In this paper, we use T_{mn} to describe the matrix of target with size $(m \times n)$,use $\overline{T}$ to describe the mean value of T_{mn}, use C_{mn} to describe the matrix of a candidate and $\overline{C}$ is its mean value. According to the sampling method, C_{mn} has the same size with T_{mn}. The the measure of structure and intensity match D_r via the Pearson Correlation Coefficient is defined by:

$$D_r = \frac{\sum_m \sum_n \left(T_{mn} - \overline{T}\right)\left(C_{mn} - \overline{C}\right)}{\sqrt{\left(\sum_m \sum_n \left(T_{mn} - \overline{T}\right)^2\right)\left(\sum_m \sum_n \left(C_{mn} - \overline{C}\right)^2\right)}} \tag{8}$$

In Eq. 8, the candidate with bigger value has more probability to be the target region.

3.5 Update the target template

To satisfy the gradually appearance changes of target object, the target template is updated according to the tracked results. We define an update factor C_t on frame t to evaluate whether to implement update by:

$$C_t = \gamma_1 \times D_c(1, t) + \gamma_2 \times D(t, t{-}1) \tag{9}$$

where $D_c(1, t)$ describes the color feature between the target regions on frame$t$ and the first frame, and it is computed by Eq. 6. $D(t, t-1)$ is the match measure between the target regions on frame$t$ and frame $t-1$, and it is calculated by Eq. 4. Bigger value of C_t means that the appearance of target on frame t has smaller variation from the target on the first frame and bigger match measure with the target on frame $t-1$. The coefficients of γ_1 and γ_2 are separately set to be 0.3 and 0.7 via the demonstrations of many experiments.

We achieve the update on frame t according to the update factor by the following three steps. First, we separately compute the update factors on frame $t-1$ and frame t. Then, we calculate the distance between the above two update factors. Third, we compare the distance with a threshold θ to decide whether to update. In this paper, we set θ to be 0.01 based on the experiments. If the difference is smaller than θ, it means the adjacent target regions on the two frames are very similar, and they both can effectively represent the initialized target object. In this case, we update target template directly by replacing the multi-features with the ones from the new tracked result on frame t. Otherwise, we preserve the multi-features unchanged as the new target template.

If we use gt1 to describe the ground truth of frame1, and use OERi to describe the OER of frame i, we can construct the pseudocode of our tracker by the following Table 1.

4 Experiments and evaluations

This paper demonstrates the effectiveness and efficiency of our method on the popular Visual Tracking Benchmark [36] which is widely used to evaluate different trackers. Our method is implemented by using Matlab R2014a (64bit) on a PC with an Intel(R) Core(TM) @2.5GHz 2.5GHz processor, RAM 16GB DDR3 memory on Windows 8.1 version. Eleven present trackers which are recently proposed and widely used in tracker evaluation are used to do comparison. These trackers include the CSK [11], CT [39], CXT [6], DFT [29], LOT [26], LSK [19], Struck [10], VTD [16], VTS [15], DLSSVM [25] and TRA [2]. 25 videos of the Benchmark are used to do experiments which covers ten challenges including the occlusion (OCC), deformation (DEF), fast move (FM), motion blur (MB), background clutter (BC), illumination variation (IV), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV) and scale variation (SV).

As shown in the recent work [36], the precision plot and success plot are used to do the quantitative evaluations of trackers. The success plot describes the percentage of successfully

Table 1 The pseudocode of our proposed method

```
/* Initialize the target template
Input (video,gt1,n=length(video));
compute the ROIs of frame1;
OER1=min(D_sim(ROIs,gt1));
Initialize template T based OER1;
for i = 2 : n
  compute the OERi of framei;
  candidates=sample(OERi);
/* compute the tracking result Resulti of framei
   for j = 1 : number(candidates)
      compute Dc(j), Ds(j), Dr(j);
           D(j) = f(Dc(j),Ds(j),Dr(j));
   end
Resulti= min(D(1),D(2),…,D(number(candidates)));
/* update target template
  compute Ct-1, Ct;
  θ=distance(Ct,Ct-1);
    if θ< 0.01
    update Template T based Resulti;
    else no update;
end
```

tracked frames by the Intersection Over Union (IOU). Bigger value of success plot reflects the better result. We calculate the IOU by the ratio between the intersection and the union of a tracked box and ground truth. When the IOU of a frame is bigger than 0.5, we denote that the tracked result on this frame is successful. We compute the precision plot by the percentage of successfully tracked frames based on the Center Location Error (CLE) with a given threshold TC ($TC = 20$ in our experiments). Bigger precision plot means more accurate results. In addition, we do the one-pass evaluation (OPE) in tracking. For more details about CLE, IOU and OPE, please review the work proposed in benchmark [36]. More evaluation results are demonstrated by Appendix 1 Figs. 11, 12, Appendix 2 Figs. 13, 14, 15, 16 (include the comparisons of total 25 videos).

4.1 Qualitative evaluations

According to the experiments, the proposed method performs very well in dealing with five challenges including occlusion, deformation, background clutter, motion blur and out-of-plane rotation. In this section, we mainly analyze the performances of our tracker on these five challenges. For the other challenges, our tracker is not the best one among the evaluated ten trackers, but it is mostly ranked as the top three trackers as demonstrated on section 4.2.

For occlusion challenge In occlusion, the target is occluded by some background. As shown in the first row of Fig. 6, the human is partly occluded by a tree. Most of compared trackers (as the Struck, VTD and VTS) introduce failures especially when the human reappears from the occlusion. Recently, people propose many trackers to deal with occlusions. For example, the DLSSVM uses the dual linear structured SVM and the RPT utilizes irregular patches. However, they still cannot produce favorable results especially in heavy occlusion as shown in Fig. 7. Our tracker can successfully reidentify the reappeared target from the heavy occlusion such as the girl and the runner in Fig. 7. That is because our update algorithm preserving the main features of the occluded target and our sampling method produce efficient candidates (Fig. 8).

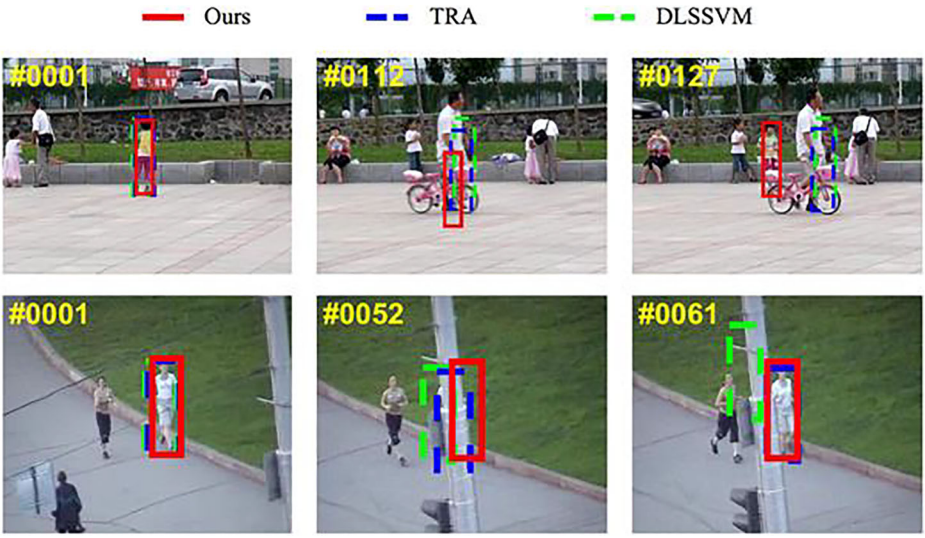

Fig. 6 The comparisons of different trackers with good performances on 5 videos

Fig. 7 The performances on heavy occlusion

For background clutter challenge This challenge often introduces the tracking drift away from the ideal target region for the interference of background. For example, the background in the second row of Fig. 6 is very cluttered and quickly changed. The results of some popular trackers such as the CXT and LOT are drifted from the right target region. Our result is more accurate as shown by the red rectangle. The main reason is we use ROI to get candidates which efficiently reduces the disturbs of cluttered background by rejecting the invalid candidates with lower salient values for owning many backgrounds.

For out-of-plane rotation challenge This challenge often brings great changes of target appearance such as the bolt in the third row and the fifth row of Fig. 6. Many trackers deal with it by quickly updating target template. However, it is disturbed by the surrounding background of target such as the advertising board in the fifth row of Fig. 6. Therefore, many trackers take the background as target such as the black and green rectangles in the third row and the fifth row of Fig. 6. Our tracker gradually updates the target template by defining an update factor to measure the appearance Figs. 9 and 10 separately describes the trends of precision plot and success plot about the 10 trackers. They reflect the average accuracy and robust of the results for all the trackers. The one on the first row and first column shows the trend of all the videos. The other five subfigures show the average trends about the videos owning the challenges of the OCC, DEF, BC, MB and OPR. According to Fig. 9, our method is the most accurate tracker among the ten trackers for owning the biggest values

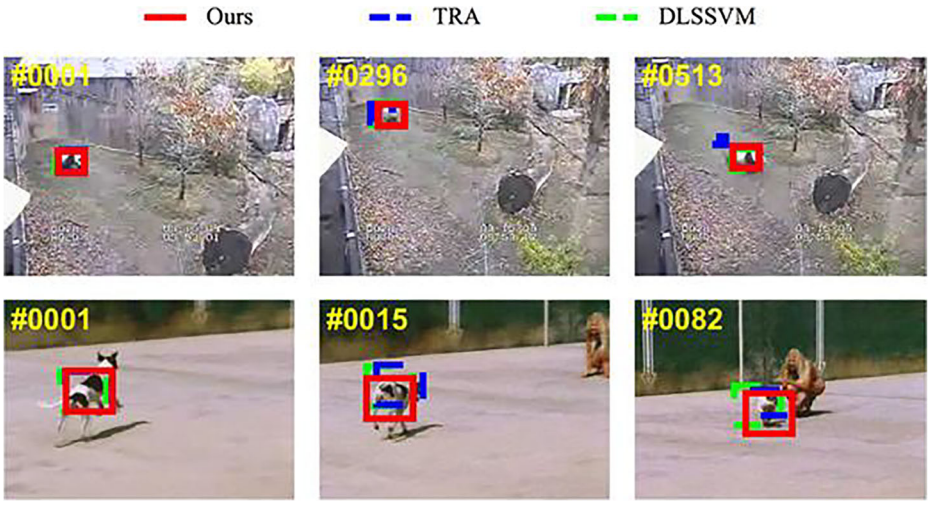

Fig. 8 The performances on drastic deformation

in these challenges. Similarly, according to Fig. 10, our method is the most robust tracker for owning the biggest Success plot.changes between adjacent target results. This update scheme preserves the primary features of target when it undergoes out of plane rotation which finally makes our tracker more robust and accurate.

For deformation challenge This challenge usually introduces great changes of target appearance representation. For example, the deformation in the second row of Fig. 6 leads the white region (the mouth of tiger) turning from left to right. Similarly, the deformation on the panda and dog in Fig. 8 makes the appearance of target object undergoes drastic variation. Many present trackers such as the CSK and TRA introduce tracking drift in dealing with deformation challenge. Our tracker produces much more favorable results than the present trackers in Fig. 6 and Fig. 8. The main reason is we use the color histogram to construct our matching method which are not influenced by shape changes or the layout variation of target appearance.

For motion blur challenge Motion blur often makes the differences between target and background very week by introducing many noises. The features of target and background are very similar which finally lead some trackers mistaking the background as target such as the tiger in the second row of Fig. 6 and the girl in the third row of Fig. 6. The tracking failures of these trackers such as the Struck and VTS come from the imprecise and unstable ability to distinguish target and background. Our method overcomes it by using three kinds of to construct target template which can efficiently reduce the disturbs of moving blurs.

4.2 Quantitative evaluations

We achieve the quantitative evaluations by Tables 2, 3, Fig. 9 and Fig. 10. Tables 2 and 3 show the precision plot and success plot of ten trackers on 25 videos. For the two evaluations, the bigger value means better tracker. In the two tables, the red, green and blue words separately describe the best, second and third tracker.

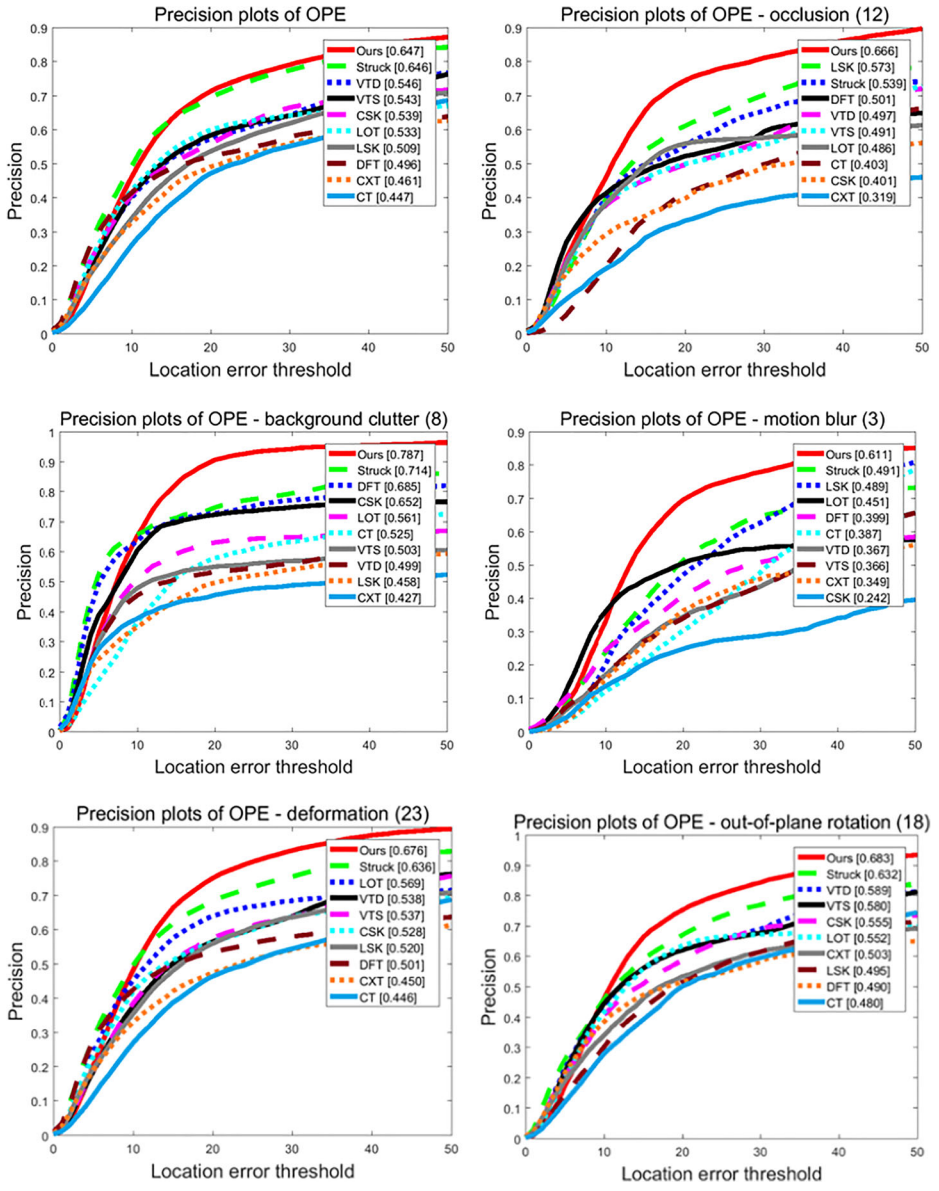

Fig. 9 The evaluations on Precision plot of some challenges that our method has favorable performance

According to Tables 2 and 3, our method performs well in many challenges. For the challenges of BC, MB, DEF, OCC and OPR, our method produced the best results for owning the biggest precision plot and success plot. It means that our method not only produces accurate results but also performs very robust. For dealing with the challenges of FM, IPR, OV and SV, our method performed as the second or the third tracker. The reason is our update scheme gradually adapts to the changes of target appearance which are common in these challenges. For the challenge of IV, the CSK produces favorable results because it efficiently learns the changes of surrounding backgrounds in tracking. The Struck tracker produces good

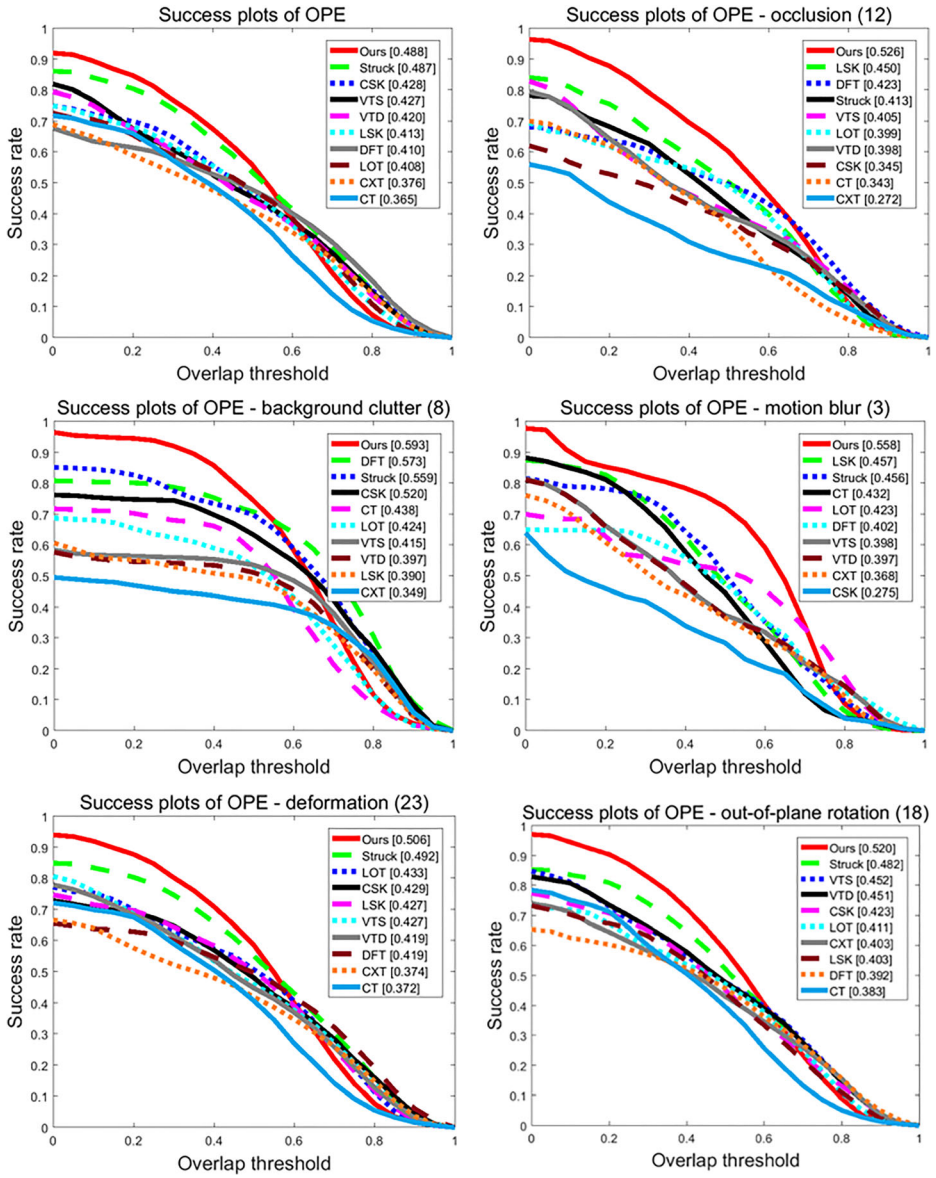

Fig. 10 The evaluations on Success plot of some challenges that our method has favorable performance

results especially for FM and IPR because it effectively deals with the noises introduced by them. Totally, the proposed tracker performs as the best tracker among all 25 videos as shown in the column of "ALL" in Tables 2 and 3.

Figures 9 and 10 separately describes the trends of precision plot and success plot about the 10 trackers. They reflect the average accuracy and robust of the results for all the trackers. The one on the first row and first column shows the trend of all the videos. The other five subfigures show the average trends about the videos owning the challenges of the OCC, DEF, BC, MB and OPR. According to Fig. 9, our method is

Table 2 The Precision Plot of the Ten Trackers on 25 Videos with Ten Challenges

Trackers	*ALL*	*FM*	*BC*	*MB*	*DEF*	*IV*	*IPR*	*OCC*	*OPR*	*OV*	*SV*
Ours	0.647	0.622	0.787	0.611	0.676	0.494	0.639	0.666	0.683	0.648	0.577
Struck	0.646	0.707	0.714	0.491	0.636	0.584	0.641	0.539	0.632	0.740	0.678
VTD	0.546	0.434	0.499	0.367	0.538	0.536	0.581	0.497	0.589	0.519	0.580
VTS	0.543	0.432	0.503	0.366	0.537	0.545	0.566	0.491	0.580	0.522	0.578
CSK	0.539	0.514	0.652	0.242	0.528	0.598	0.503	0.401	0.555	0.352	0.516
LOT	0.533	0.440	0.561	0.451	0.569	0.343	0.517	0.486	0.552	0.241	0.552
LSK	0.509	0.454	0.458	0.489	0.520	0.470	0.547	0.573	0.495	0.443	0.456
DFT	0.496	0.458	0.685	0.399	0.501	0.573	0.447	0.501	0.490	0.623	0.373
CXT	0.461	0.509	0.427	0.349	0.450	0.399	0.491	0.319	0.503	0.499	0.533
CT	0.447	0.446	0.525	0.387	0.446	0.358	0.394	0.403	0.480	0.577	0.475

(Red, green and blue separately means the best, the second and the third)

the most accurate tracker among the ten trackers for owning the biggest values in these challenges. Similarly, according to Fig. 10, our method is the most robust tracker for owning the biggest Success plot.

4.3 Implement efficiency

The frames processed per second (fps) is often used to describe the efficiency of trackers. Only with the big value of fps, the tracker can support the online process. Table 4 describes the fps of all the 12 trackers. The CSK and TRA methods track target very fast for owning 362 fps and 282 fps. The reason is they utilize the circled matrix to achieve tracking. Our tracker produces 137.6 frames per second which can achieve online tracking. The reason is our methods to match and update target template is very fast and robust.

Table 3 The Success Plot of the Ten Trackers on 25 Videos with Ten Challenges

Trackers	*ALL*	*FM*	*BC*	*MB*	*DEF*	*IV*	*IPR*	*OCC*	*OPR*	*OV*	*SV*
Ours	0.488	0.528	0.593	0.558	0.506	0.432	0.505	0.526	0.520	0.529	0.436
Struck	0.487	0.602	0.559	0.456	0.492	0.457	0.523	0.413	0.482	0.594	0.484
VTD	0.420	0.403	0.397	0.397	0.419	0.430	0.478	0.398	0.451	0.458	0.431
VTS	0.427	0.416	0.415	0.398	0.427	0.449	0.469	0.405	0.452	0.462	0.436
CSK	0.428	0.456	0.520	0.275	0.429	0.494	0.431	0.345	0.423	0.338	0.390
LOT	0.408	0.374	0.424	0.423	0.433	0.301	0.389	0.399	0.411	0.235	0.416
LSK	0.413	0.417	0.390	0.457	0.427	0.388	0.458	0.450	0.403	0.395	0.370
DFT	0.410	0.436	0.573	0.402	0.419	0.494	0.391	0.423	0.392	0.544	0.302

(Red, green and blue separately means the best, the second and the third)

Table 4 The Numbers of Frames Processed Per Second (fps) with Different Trackers

Tracker	Ours	Struck	VTD	VTS	CSK	LOT	LSK	DFT	CXT	CT	TRA	DLSSVM
FPS	137.6	20.2	5.7	5.7	362	0.7	5.5	13.2	15.3	64.4	282	25

5 Conclusion

This paper proposes a robust and fast visual tracking method by identifying and matching target template. Our method constructs target template by three kinds of features related to the color feature, shape and contour feature, the features about the distributions of structure and intensity. A new method to compute the ROIs is defined which improving the efficiency of candidates by providing accurate sampling region and reducing redundant samples. Many experiments have done using 25videos with 10 kinds of tracking challenges from the tracking benchmark 2013. By the quantitative and qualitative evaluations, our tracker performs more favorable than the famous present 11 trackers especially in the challenges of BC, MB, DEF, OCC and OPR. As our target template cannot reduce the noise from low resolution, our tracker sometimes fails in tracking target shot in low resolution and with fast movement. In the future, we will introduce more features in the target template to overcome the challenge from low resolution and motion blur from fast target movement.

Acknowledgments This paper is funded by some projects of the authors. Yun Liang is the Natural Science Foundation of China (No. 61772209), and the Science and Technology Planning Project of Guangdong Province (No. 2019A050510034, 2019B020219001). Jian Zhang is supported by the Natural Science Foundation of China (No. 61972361). Chen Lin is supported by the Natural Science Foundation of China (No.61472335, 61,972,328). Jun Xiao is supported by the National Natural Science Foundation of China (No.61572431), and the Zhejiang Natural Science Foundation (No.LY17F020009, LR19F020002, LZ17F020001).

Appendix 1: All The Precision Plots and Success Plots

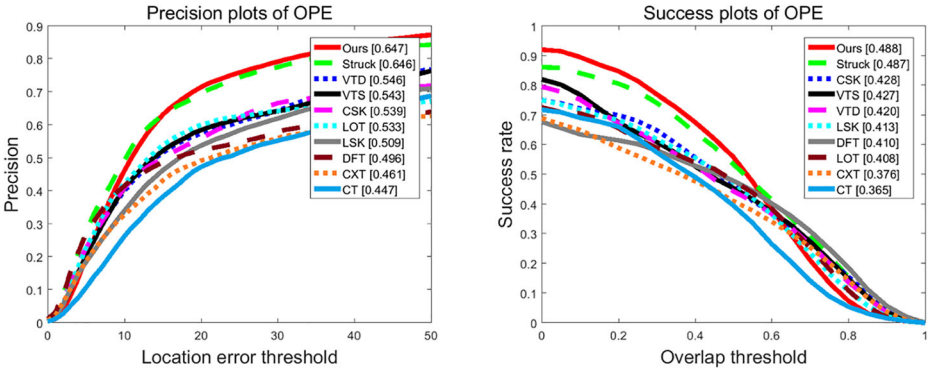

Fig. 11 The total precision plots and success plots

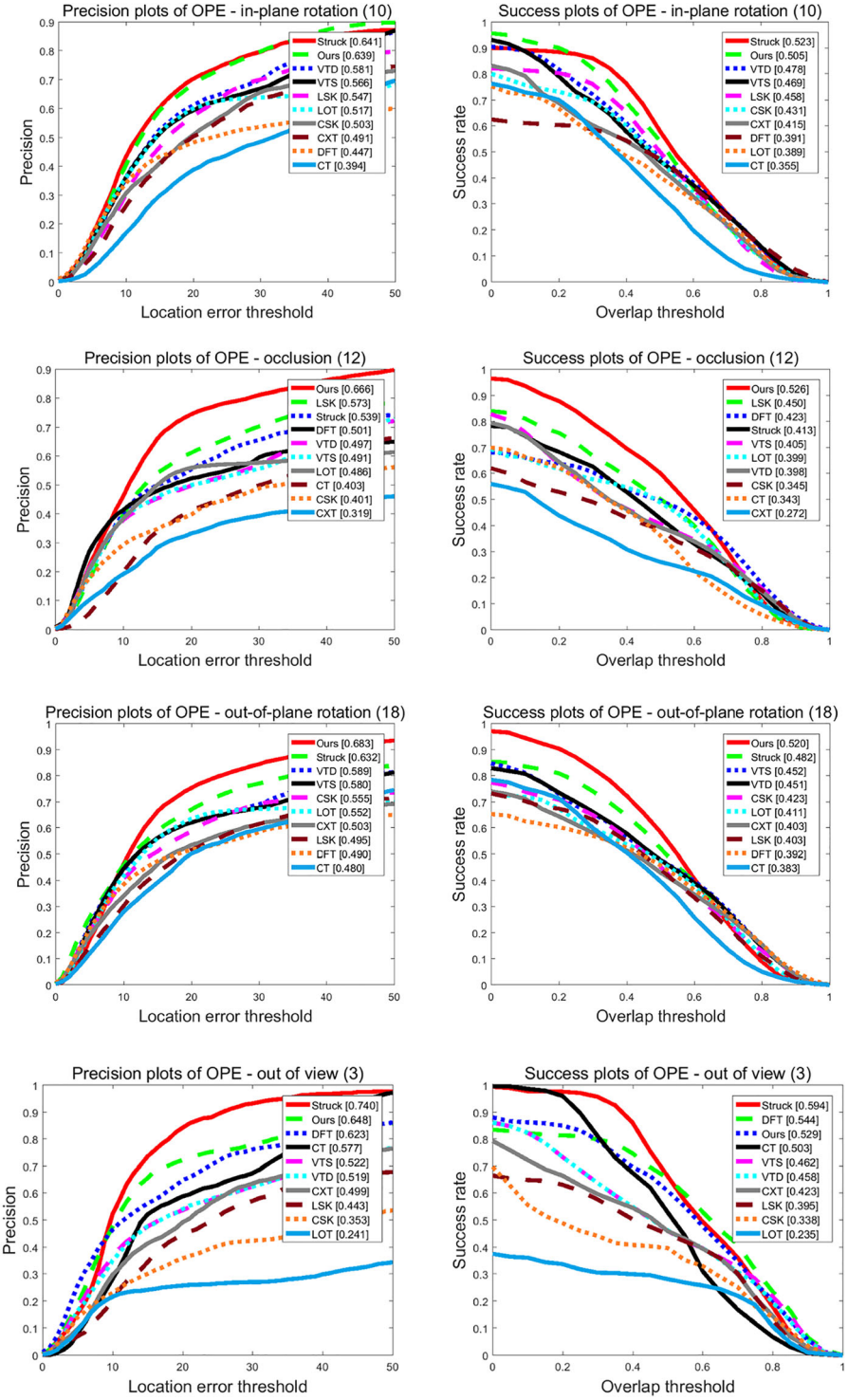

Fig. 12 The total precision plots and success plots

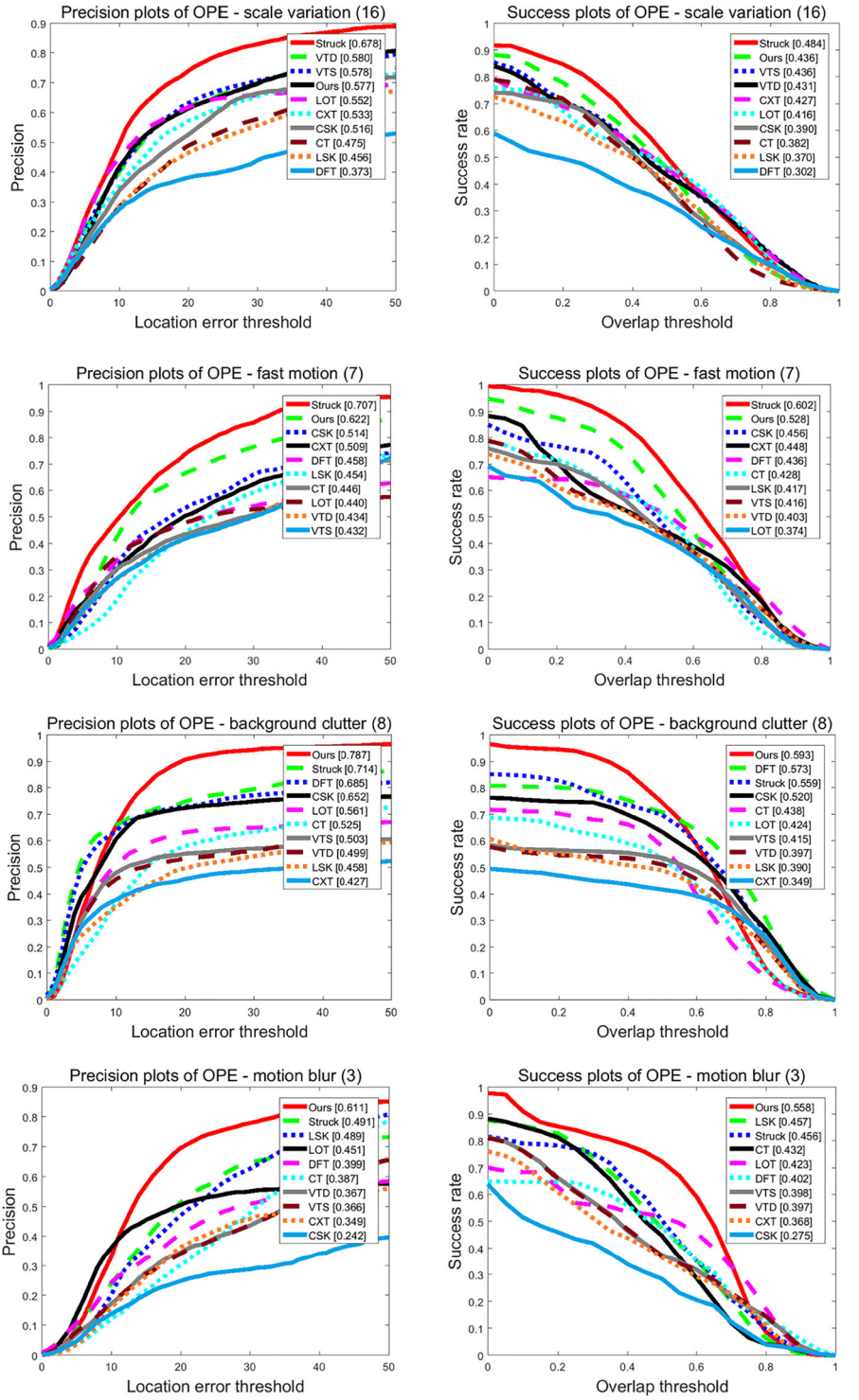

Fig. 12 (continued)

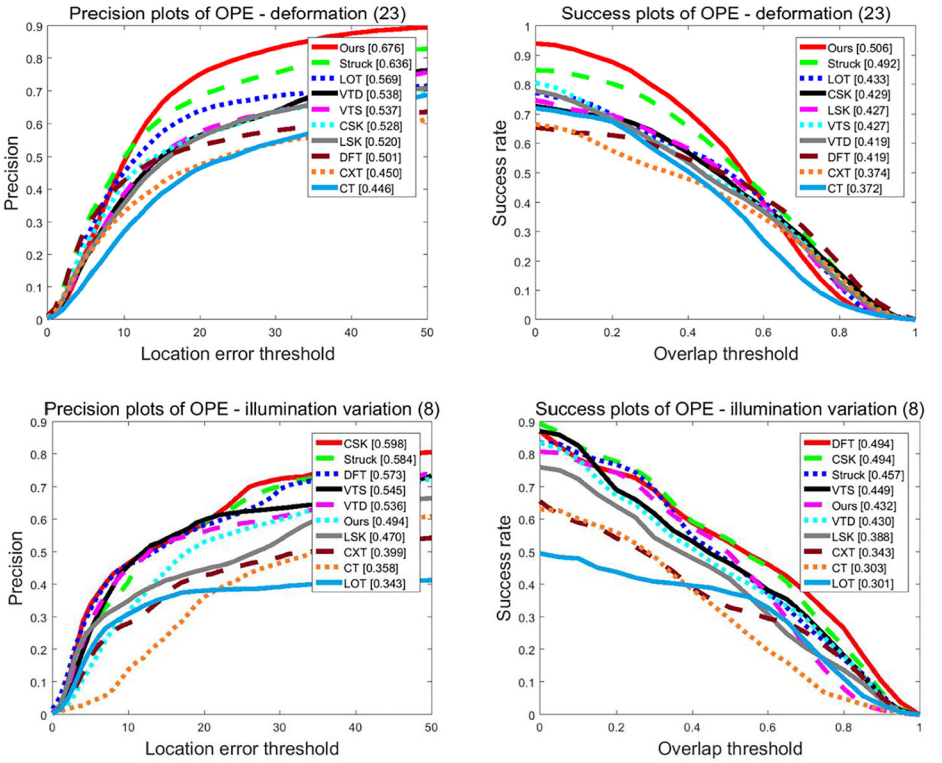

Fig. 12 (contin\ued)

Appendix 2 The comparisons of total 25 videos

Fig. 13 comparisons of eight videos.

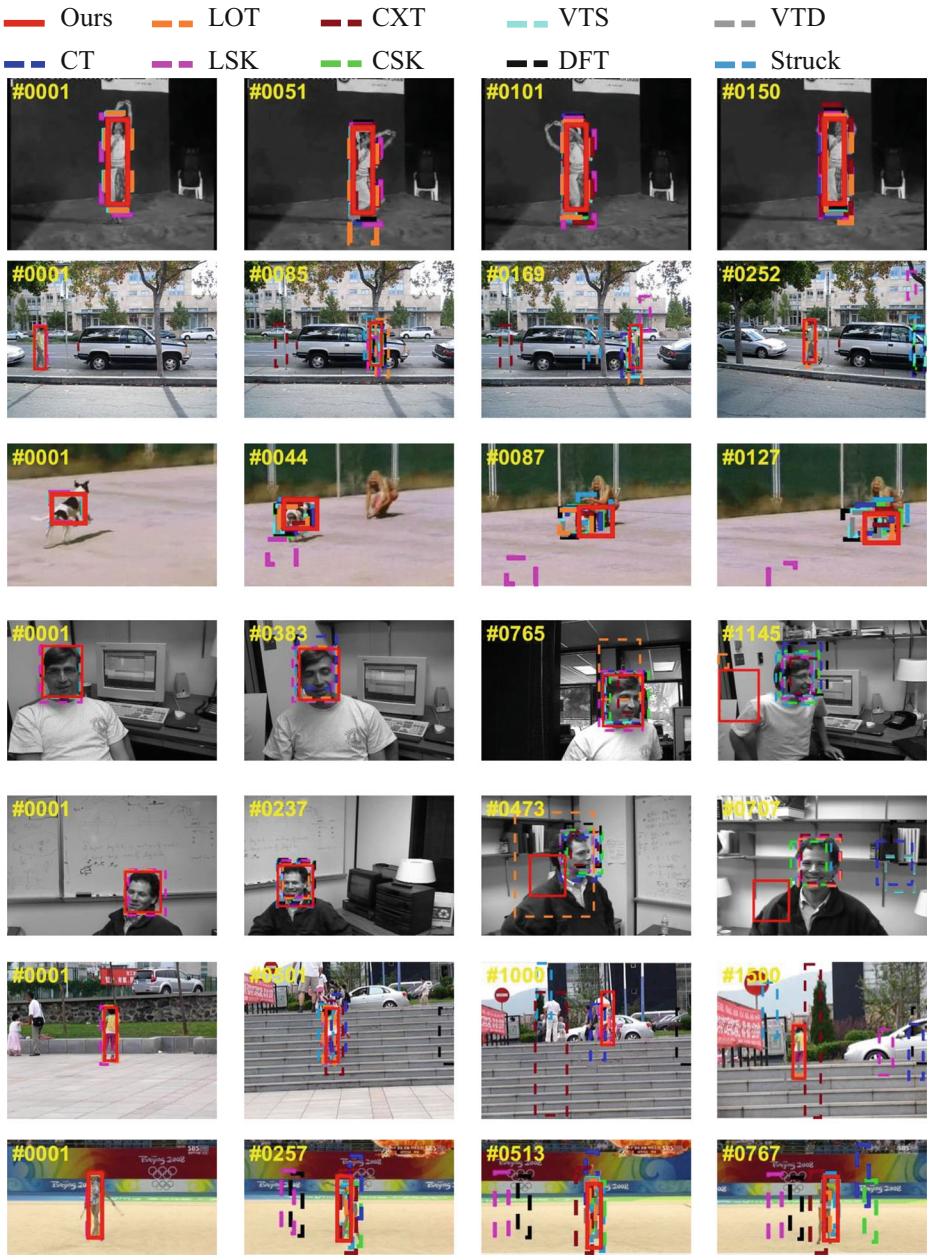

Fig. 14 Comparisons of seven videos

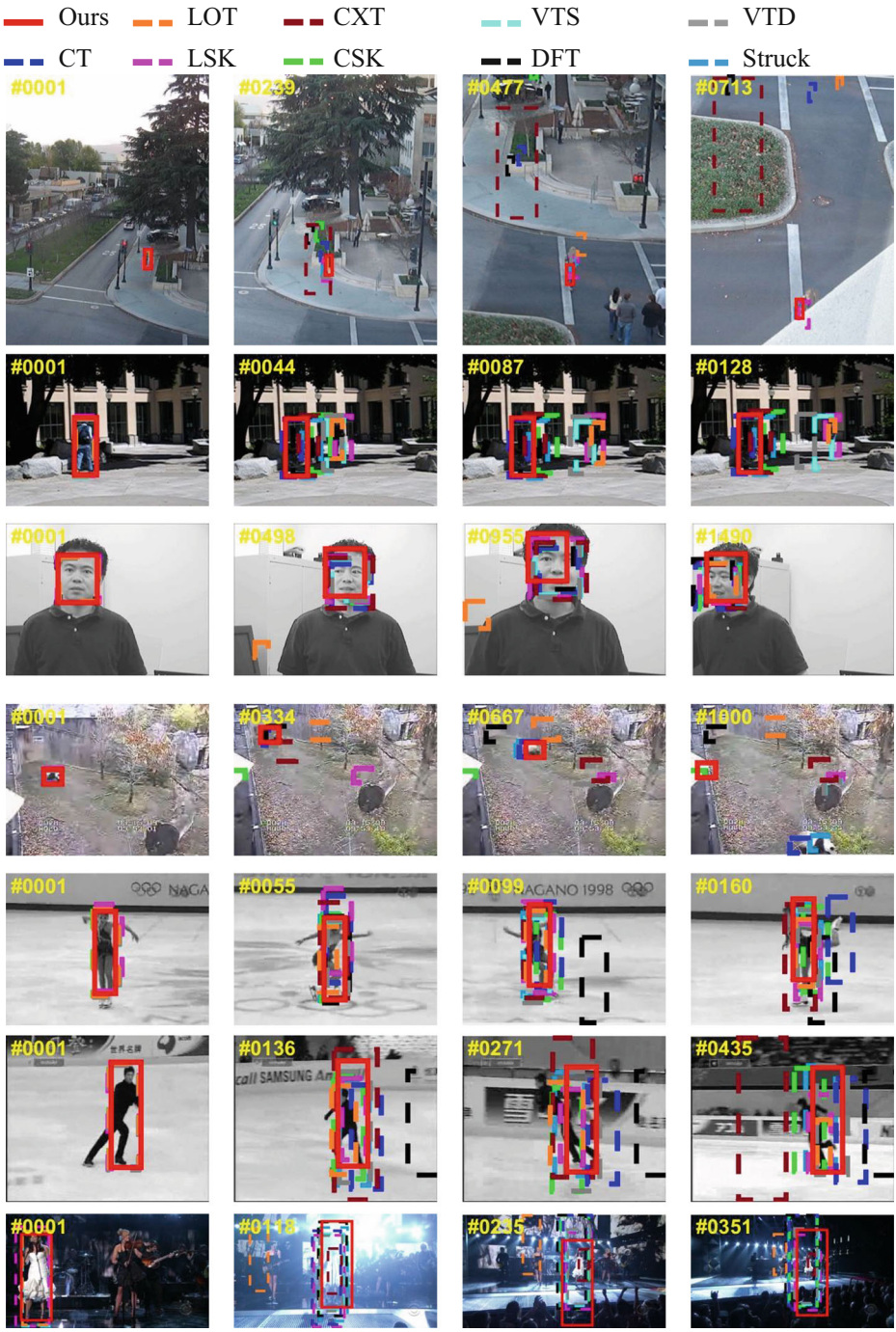

Fig. 15 Comparisons of seven videos.

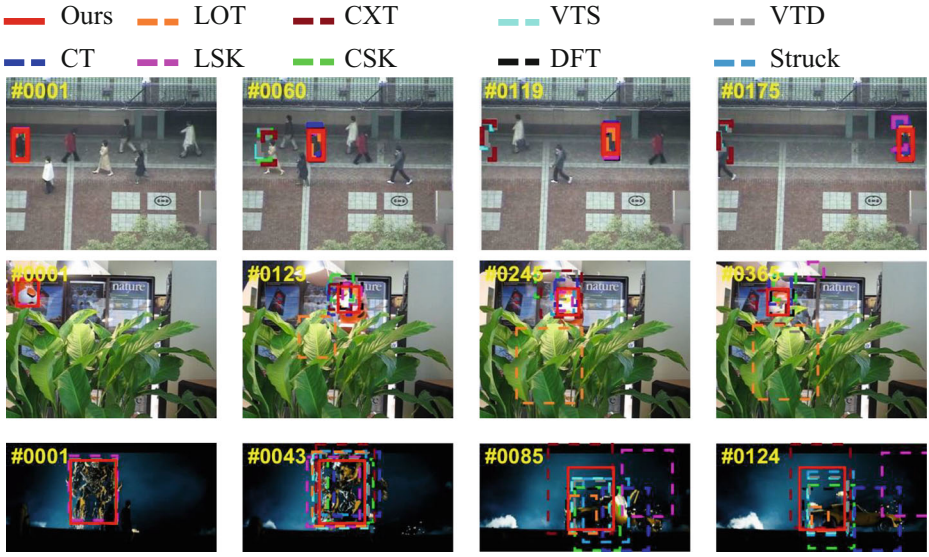

Fig. 16 Comparisons of three videos

References

1. Adam A, Rivlin E, Shimshoni I (2006) Robust fragments -based tracking using the integral histogram. IEEE conference on computer vision and pattern recognition, p.798-805
2. Bibi A, Mueller M, Ghanem B (2016) Target response adaptation for correlation filter tracking. European conference on computer vision, p.419-433. 2
3. Bolme DS, Beveridge JR, Draper BA, et al. (2010) Visual object tracking using adaptive correlation filters. IEEE conference on computer vision and pattern recognition, p.2544–2550
4. Cai Z, Wen L, Yang J, et al. (2012) Structured visual tracking with dynamic graph. Asian Conference on Computer Vision. Springer Berlin Heidelberg, p.86–97
5. Choi J, Chang HJ, Fischer T, Yun S, Lee K, Jeong J, Demiris Y, Choi JY (2018) Context-Aware Deep Feature Compression for High-Speed Visual Tracking. IEEE conference on computer vision and pattern recognition, p.479–488
6. Dinh TB, Vo N, and Medioni G (2011) Context Tracker: Exploring supporters and distracters in unconstrained environments. IEEE conference on computer vision and pattern recognition, p.1177–1184
7. Godec M, Roth PM, Bischof H (2013) Hough-based tracking of non-rigid objects. Comput Vis Image Underst 117(10):1245–1256
8. Grabner M, Grabner H, Bischof H (2007) Learning features for tracking. IEEE conference on computer vision and pattern recognition, p.1–8
9. Guo Y, Chen Y, Tang F, Li A, Luo W, Liu M (2014) Object tracking using Learned feature manifolds. Comput Vis Image Underst 118:128–139
10. Hare S, Saffari A, and Torr PHS (2011) Struck: structured output tracking with kernels. International conference on computer vision, p.2096–2109
11. Henriques JF, Caseiro R, Martins P, et al. (2012) Exploiting the circulant structure of tracking-by-detection with kernels. European conference on computer vision. Springer Berlin Heidelberg, p. 702–715
12. Hinterstoisser S, Lepetit V, Ilic S, et al. (2010) Dominant orientation templates for real-time detection of texture-less objects. IEEE conference on computer vision and pattern recognition 10, p.2257–2264
13. Hong Z, Chen Z, Wang C, Mei X, Prokhorov D, and Tao D. (2015) Multi-store tracker (MUSTer): a cognitive psychology inspired approach to object tracking. IEEE conference on computer vision and pattern recognition, 2015, p. 749–758
14. Jia X, Lu H, Yang MH (2012) Visual tracking via adaptive structural local sparse appearance model. IEEE conference on computer vision and pattern recognition, p.1822–1829
15. Kwon J, Lee KM (2010) Visual tracking decomposition. IEEE conference on computer vision and pattern recognition, p1269–1276

16. Kwon J, Lee KM (2011) Tracking by sampling trackers. International conference on computer vision, p.1195–1202
17. Kwon J, Lee KM (2013) Highly nonrigid object tracking via patch-based dynamic appearance modeling. IEEE Trans Pattern Anal Mach Intell 35(10):2427–2441
18. Li F, Tian C, Zou W, Zhang L, Yang M-H (2018) Learning spatial-temporal regularized correlation filters for visual tracking. IEEE conference on computer vision and pattern recognition
19. Liu B, Huang J, Yang L, and Kulikowsk C (2011) Robust tracking using local sparse appearance model and K-selection. IEEE conference on computer vision and pattern recognition, p.1313–1320
20. Liu AA, Su YT, Nie WZ, Kankanhalli M (2017) Hierarchical clustering multi-task learning for joint human action grouping and recognition. IEEE Trans Pattern Anal Mach Intell 39(1):102–114
21. Liu AA, Xu N, Zhang H et al. (2018) Multi-level policy and reward reinforcement learning for image captioning, twenty-seventh international joint conference on artificial intelligence (IJCAI), p.821-827
22. Liu L, Yan X, Shen A (2019) Adaptive multi-feature fusion for correlation filter tracking. Commun Signal Process Syst 1:1057–1066
23. Ma C, Liu C, Peng F, Liu J (2016) Multi-feature hashing tracking. Pattern Lett 69:62–71
24. Maresca ME, Petrosino A, Matrioska (2013) A multi-level approach to fast tracking by learning. International Conference on Image Analysis and Processing. Springer Berlin Heidelberg, p.419–428
25. Ning J, Yang J, Jiang S, Zhang L and Yang MH (2016) Object tracking via dual linear structured SVM and explicit feature map. IEEE conference on computer vision and pattern recognition, p.4266-4274
26. Oron S, Bar-Hillel A, Levi D, Avidan S (2012) Locally orderless tracking. IEEE conference on computer vision and pattern recognition, p.1940–1947
27. Pearson K, Galton F (2014) Pearson product-moment correlation coefficient. Covariance
28. Ross DA, Lim J, Lin RS, Yang MH (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1–3):125–141
29. Sevilla-Lara L and Learned-Miller E (2012) Distribution fields for tracking. IEEE conference on computer vision and pattern recognition, p.1910–1917
30. Song Y, Ma C, Gong L, Zhang J, Lau RWH, Yang M-H (2017) CREST: convolutional residual learning for visual tracking. International conference on computer vision, p.2555–2563
31. Stadler S, Grabner H, Van Gool L (2012) Dynamic objectness for adaptive tracking. Conference Asian Conference on Computer Vision, p43–56
32. Tu WC, He S, Yang Q, Chien SY (2016) Real-time salient object detection with a minimum spanning tree. IEEE conference on computer vision and pattern recognition, p.2334–2342
33. Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PHS (2017) End-to-end representation learning for correlation filter based tracking. IEEE conference on computer vision and pattern recognition, p.5000-5008
34. Vojíř T, Matas J (2014) The enhanced flock of trackers. Registration and Recognition in Images and Videos. Springer Berlin Heidelberg, p.113–136
35. Wang D, Lu H, Yang MH (2013) Least soft-threshold squares tracking. IEEE conference on computer vision and pattern recognition. P.2371–2378
36. Wu Y, Lim J, Yang M H (2013) Online object tracking: a benchmark. IEEE conference on computer vision and pattern recognition, p.2411–2418
37. Xu N, Liu AA, Wong Y, Zhang Y, Nie WZ, Su YT, Kankanhalli M (2019) Dual-stream recurrent neural network for video captioning. IEEE Trans Circuits Syst Video Technology 29(8), p.2482–2493
38. Zhang L, Van Der Maaten L (2014) Preserving structure in model-free tracking. IEEE Trans Pattern Anal Mach Intell 36(4):756–769
39. Zhang K, Zhang L, and Yang MH (2012) Real-time compressive tracking. European Conference on Computer Vision, p864–877
40. Zhang J, Yu J, Tao D (2018) Local deep-feature alignment for unsupervised dimension reduction. IEEE transactions on image processing, p. 1-1
41. Zhu G, Wang J, Wu Y, et al. (2016) MC-HOG correlation tracking with saliency proposal. Proceedings of the thirtieth AAAI conference on artificial intelligence. AAAI press, p. 3690-3696.
42. Zhu G, Porikli F, Li H (2016) Beyond local search: tracking objects everywhere with instance -specific proposals. IEEE conference on computer vision and pattern recognition, p.943–951

Yun Liang is an Associate Professor at the College of Mathematics and Informatics, South China Agriculture University, Guangzhou, China. She received her M.Sc and Ph.D degree in the School of Information Science and Technology at Sun Yat-sen University separately in 2005 and 2011. From 2016 to 2017, she worked in the Simon Fraser University cooperated with Professor Ping Tan. Yun Liang's research interests include computer vision, image computation, machine learning, etc.

Jian Zhang received his B.E. degree and M.E. degree from Shandong University of Science and Technology in 2000 and 2003 respectively, and received the Ph.D. degree from Zhejiang University in 2008. He was a postdoctoral researcher in the department of mathematics of Zhejiang University from 2009 to 2011. Since 2011, he has been working in School of Science & Technology of Zhejiang International Studies University. In 2016, he worked as a visiting scholar in School of Computing Science of Simon Fraser University for six months. His research interests include deep learning, multimedia processing and machine learning. He serves as a reviewer of several prestigious journals and was awarded the certificates of outstanding contribution in reviewing by the Journal of Pattern Recognition and Neurocomputing.

Meihua Wang is an Associate Professor and Master adviser of college of Mathematics and Informatics in South China Agricultural University. She received her M.Sc degree from South China University of Technology in 1999. She has finished two big international cooperative projects with ADSC(Advanced digital Science Central) since 2010. Her research interest includes image processing, computer vision and machine learning.

Chen Lin received a B. Eng. degree and a Ph.D. degree both from Fudan University, China in 2004 and 2010. She is currently an Associated Professor at School of Information Science & Technology, Xiamen University, China. Her research interests include web mining and recommender systems.

Jun Xiao received the Ph.D. degree in computer science and technology from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2007. He is currently a Professor with the College of Computer Science, Zhejiang University. His current research interests include computer animation, multimedia retrieval, and machine learning.