Multimedia Tools and Applications (2019) 78:14195–14230
https://doi.org/10.1007/s11042-018-6760-4

Robust visual tracking via identifying multi-scale patches

Yun Liang [1] · Ke Li [2] · Jian Zhang [3] · Meihua Wang [1] · Chen Lin [4]

Received: 22 March 2018 / Revised: 26 September 2018 / Accepted: 4 October 2018 /
Published online: 1 November 2018

Abstract

The complex changes of target and its surroundings introduce several tracking challenges, such as occlusion, deformation and so on. Many challenges coexist in a video which makes tracking still under successfully solved. The present trackers deal with coexisting challenges in a common model for all components of target. However, different components often undergo different challenges at the same time, while some with deformation and others with occlusion. The common model cannot adapt to these challenges simultaneously. An effective method is to separately deal with the challenges. This paper proposes a new robust tracker via separately tracking and identifying the multi-scale patches of target to cope with the coexisting challenges. It is achieved by three respects. Firstly, we define a new basic tracker by introducing the gaussian mixture model into Kernelized Correlation Filters (KCF). For the KCF is very sensitive to the similar surroundings, we construct a regular term and a loss function via the gaussian mixture model to optimize the classifier formed by KCF. Secondly, we define a new appearance representation model of target by multi-scale patches. To deal with the different variations of patches, we separately construct and update their appearance representations. Thirdly, with the tracked result of each patch computed by our basic tracker, we use the structure information and the Hough Vote to decide the target. Then, our method improves the accuracy by rejecting the failed tracked patches. Many experiments have been achieved on the Tracking Benchmark, and the quantitative and qualitative evaluations show that the proposed tracker performs better than most of the present trackers.

Keywords Multi-scale patches · Visual tracking · Kernelized correlation filters · Gaussian mixture model · Hough vote

1 Introduction

Visual Tracking plays an important role in computer vision for its ability of identifying the moving target in videos [35, 37]. From the tracked results, people can predict the action and even the activity of moving object [25, 26], which helps intelligent devices to understand the

✉ Yun Liang
sdliangyun@163.com

Extended author information available on the last page of the article

high-level semantic information. Therefore, visual tracking has been widely used in automatic drive, video surveillance, drone navigation, virtual reality and so on. Recently, many visual tracking methods are proposed such as the methods based on deep learning [8, 10, 30, 33, 38, 45] and the methods based on correlation filters [2, 6, 7, 15, 17, 20, 27, 31]. However, the unpredictable and continuous changes of target object and its surroundings bring many tracking challenges, including illumination variation, target deformation, occlusion, motion blurs and so on. Furthermore, some challenges usually come out at the same time, which lead to the present trackers fail in producing robust results. The main reason is these trackers propose a common model to deal with the different challenges from different components of target. The common model may produce good results on one challenge but bad results on another challenge. Therefore, the tracked results often are drifted from the accurate position by the wrongly tracked components of target. The effective method is to separately deal with the challenges from different components of target. Recently, many researchers prefer to utilize local features of target computed by the correlation filters to detect an object [16, 39], which can help trackers to adapt to the appearance change of target by verifying the representations of its local regions. However, as demonstrated in the previous work [15], the kernelized correlation filters is very sensitive to the similar surroundings and usually produces more than one peaks in tracking which lead to tracking drift and failure. Furthermore, for without the structure constraint between regions and the target, the trackers [13, 44] based on local features often introduces and expands the tracking errors from target region which lead to the finally tracking failure or drifting. In addition, for the high speed and good performance in solving photometric or geometric variations, the correlation filters especially its improved version the kernelized correlation filters [15] are often used to achieve the tracking of local patches.

Therefore, this paper proposes a robust visual tracker by introducing gaussian mixture model into kernelized correlation filters to deal with the challenges from similar surroundings, and using the structure information with the separating tracking of the multi-scale patches of target. The contributions of this paper are as follows:

1) A new basic tracker is defined to track the multi-scale patches of target. This tracker is defined by introducing gaussian mixture model into the kernelized correlation filters. A new regular term and loss function is constructed to optimize the classifier from kernelized correlation filters to form our tracker. This tracker effectively deal with the simultaneously emerging peaks which leads to tracking failure and are introduced by the surroundings of target.
2) A new appearance representation of target is proposed based on multi-scale patches. To satisfy the complex changes brought by the different variations of local target regions, we separately construct and update the appearance representations of multi-scale patches to get the appearance representation of target. This method successfully and flexibly represents the challenging appearance change of target by differently updating its representations of patches.
3) The structural information between patches and target is defined and updated to computing the final tracking results. After using our new basic tracker to track all the patches of target, we employ the structural information and Hough Vote to compute target region. The structural information describes the relative layouts between patches and target, and rejects the failed tracked patches to take part in detecting target to improve the tracking accuracy. It is updated in tracking to satisfy the relative motions and the different variations between patches and target.

In summary, the proposed tracker exploits multi-scale patches to satisfy the complex and drastic appearance changes of target, and combines the gaussian mixture model with kernelized correlation filters to cope with the challenge of similar surroundings of target. As shown in Fig. 1, our tracker produces favorable results when deals with some unpredictable and coexisting challenges. The quantitative and qualitative evaluations on the TB-50 (50 videos) in Tracking Benchmark [42], demonstrate that the proposed tracker performs better than most of the present trackers.

2 Related work

The key problem of robust visual tracking is to construct a good appearance representation model of target which can accurately describe the unpredictably changing appearance of target [37]. According to how to construct the appearance representation model of target, there are generally two kinds of tracking methods. One method is to construct the model based on the global features extracted from the whole target region [1, 5, 14, 29, 32, 34, 40, 41]. The other method is to build the model based on the set of local features computed from the local patches of target region [4, 11, 16, 18, 21, 22, 39, 44].

Tracking methods based on the global features For this kind of tracking methods, it uses the differences between the candidates of target region and the appearance representation model to predict target [1, 34]. For example, Wang et al. [40] proposed to use the circulant feature maps about target computed by correlation filters to represent the tracking object. Wang

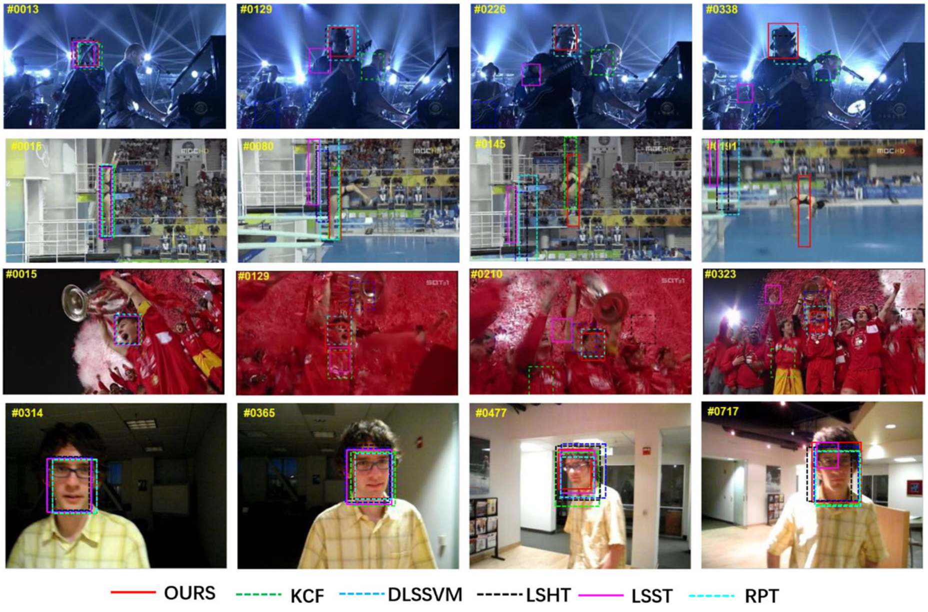


Fig. 1 The proposed tracker performs more favorable in dealing with “Shaking” (first row, with 5 challenges: IV, SV, IPR, OPR, BC), “Diving”(second row, with 3 challenges: SV, DEF, IPR), “Shocker”(third row, with 8 challenges:IV, SV, OCC, MB, FM, IPR, OPR, BC) and David (fourth row, with 7 challenges: IV, SV, OCC, DEF, MB, IPR, OPR)

et al. [41] suggested to use locality sensitive histogram of target and an adaptive Hamiltonian Monte Carlo sampling to deal with the appearance variation and abrupt motion of target. Ning et al. [34] tracked target based on the bilinear support vector machine (SVM), which improved the accuracy and reduced computing complexity by analyzing the structural SVM and extracting the global features of target. Bibi et al. [1] advised to utilize the global features from color channel and LRT channel in an improved kernelized correlation filters. Mohanapriya et al. [32] utilized the textural pattern analysis to define a novel background normalization method to suppress the shadow influences.

Recently, many trackers based on deep learning algorithms are proposed in [8, 10, 45]. They usually design a deep network to learn and control the action of target. These methods often initialize the global representation of target depending on some pre-training models and often update it in the learning and retraining process. For example, Fan et al. [8] utilized the recurrent neural network (RNN) to model object structure, and incorporated it into CNN to improve its robustness to similar distractors. Hamed et al. [10] proposed to exploit the real background patches together with the target patch to learn the tracker defined by the correlation filters. Yun et al. [45] proposed to use different sequences to achieve pre-training and tune the deep network to achieve the appearance update of target and background.

In a word, for the kind of tracking methods via global features, it usually performs well in tracking the target with rigid deformation which can preserve the global feature and structure of target undergoing very little and simple change through a video. However, it suffers from the local changes from the componets of target, and usually leads to tracking drift or failure. Recently, many people try to improve this kind of method by introducing the local features of target.

Tracking methods based on the local features For this kind of tracking method, it first divides the target object into many local patches, then uses the appearance model of local patches to represent and track target [21, 39]. This kind of method performs well when local deformation and occlusion occurs, because they can locally change the representation of target while greatly preserve the unchanged parts. Following this way, these methods can gradually and accurately adapt to the change of target, which finally lead to less tracking failures and drifts. Recently, many tracking methods based on local features are proposed [4, 16, 18, 21, 22, 39, 44]. For example, He et al. [13] suggested to represent patches by local histograms, Hare et al. [11] proposed to present target by local feature points, Yang et al. [44] advised to present target by superpixels, Wang et al. [39] suggested use patches in sparse coding scheme, Hu et al. [16] defined midlevel cues on superpixels level to describe target based on target-background saliency confidence map.

However, these present trackers usually represent target by its discrete local appearance models which lose the globally constraints and description of the whole target. Therefore, the multi-model technique is introduced to help people exploit both the global and local information. For example, Liu et al. [28] utilized the multi-source learning framework with fused lasso penalty to predict future career based on the people's social network. In visual tracking, people construct the multi-model technique by proposing the constraints between local appearance representation of target to improve the accuracy [23, 36, 46]. Li et al. [22] achieved the structural constraint between local patches of target by adjusting the relative positions between local patches and the target object. Chen et al. [4] constructed the global constraint by defining multi-scale layer representations of target, and different layer referred to the patch with different size. Jia et al. [18] utilized the structural sparse representation of patches to achieve the global constraint. Xu et al. [43] proposed to combine the Low-Dimensional and High-Dimensional approaches in one framework to deal with the challenges from various motions in tracking human action. In addition, Liu et al. [24] suggested to use the Gaussian Process Dynamical Model (GPDM) and the Annealed Particle Filtering (APF) to overcome the challenges from action tracking of human.

From the above description and analysis, we can conclude that it is an effective approach to design a tracker by using local features of target to construct its appearance model and introducing the globally constraints which reflect the layout structure between target and its local regions. Following this way, we propose a new robust tracker by identifying the multi-scale patches of target, which represents target according to the appearance and structure information of its multi-scale patches and tracks patches by defining a new basic tracker based on the gaussian mixture model and kernelized correlation filters. There are two obvious advantages of our method. First, it uses patches with different sizes and layout structure between them to represent target, which makes the tracker utilize both the local and global information of target to solve coexisting challenges in tracking. Second, it effectively reduces tracking failure by dealing with the peaks' disturbs introduced by kernelized correlation filters and the similar surroundings of target. Many experiments on different kinds of videos and tracking challenges have demonstrated that our method performs much better in dealing with tracking challenges especially when these challenges emerge simultaneously.

3 The proposed tracker

The proposed tracker is constructed by three components according to the process order, including constructing the appearance representation of target, defining the algorithm of this tracker to get the position and the size of target based on the appearance representation, and updating the appearance representation. The framework of the proposed tracker is shown as Fig. 2. First, the proposed tracker constructs the target appearance representation by randomly departing the target object into patches with multi-scale sizes and building the appearance representation for each patch by the kernelized correlation filters. Second, a new basic tracker is defined by introducing the gaussian mixture model into kernelized correlation filters to track each patch. Then, the proposed tracker computes the target position and size by the Hough vote under the global constraints of the tracked results of patches. Finally, it separately updates the appearance representation of all patches by updating the preserved patches and initializing the resampled patches to adapt to the changes of target and its surrounding background.

The overall flowchart of our proposed tracker is described by Fig.3. There are six main processes in our tracking method. First, we extract many multi-scale patches around the target region (the orange rectangle in Fig.3 (a)). Some patches cover the target region such as the pink, red, purple, light blue and cyan rectangles in Fig.3 (a). Some patches cover the

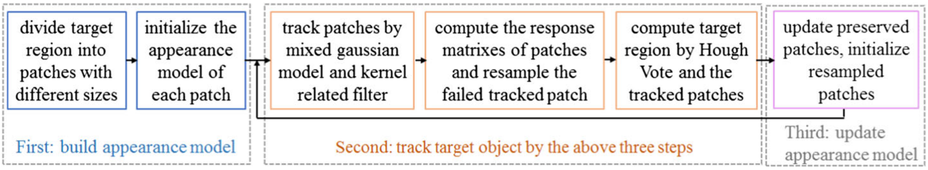

divide target region into patches with different sizes
initialize the appearance model of each patch
First: build appearance model
track patches by mixed gaussian model and kernel related filter
compute the response matrixes of patches and resample the failed tracked patch
compute target region by Hough Vote and the tracked patches
Second: track target object by the above three steps
update preserved patches, initialize resampled patches
Third: update appearance model

Fig. 2 The framework of the proposed tracker

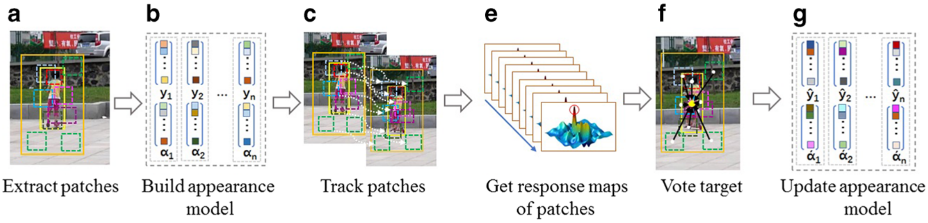

a
b
c
e
f
g
y_1 y_2 ... y_n
α_1 α_2 α_n
$\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_n$
$\dot{\alpha}_1$ $\dot{\alpha}_2$ $\dot{\alpha}_n$
Extract patches
Build appearance model
Track patches
Get response maps of patches
Vote target
Update appearance model

Fig. 3 The overall flowchart of the proposed tracker. It repeats from (c) to (g) to achieve the whole tracking of a video

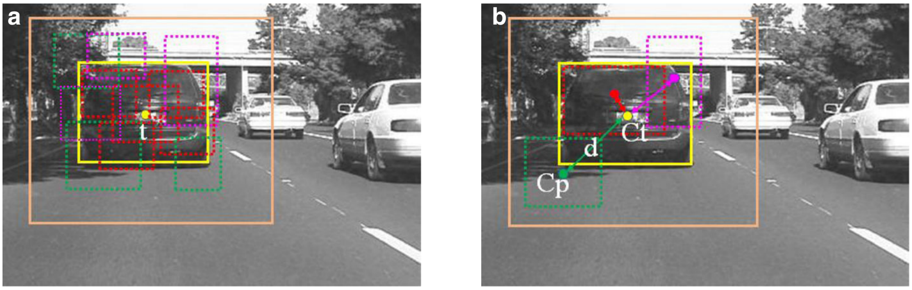
surrounding background such as the green rectangles in Fig.3 (a). Second, we use our basic tracker to construct the appearance representation of each patch by two matrix such as the (y_1, α_1) for the first patch in Fig.3 (b). Third, we use our basic tracker to track each patch on the coming frame as shown in Fig.3 (c). Fourth, for each patch, our tracker computes a response map whose highest peak is the center location of a tracked patch such as point in red circle in Fig. 3 (e). Fifth, we use the center of each tracked patch (as the white points in Fig.3) to vote the center of target object (as the yellow point in Fig. 3(f)). Finally, our tracker updates the appearance representation of each patch by updating its two matrixes such as the matrixes $(\hat{y}_1, \alpha_1)$ for the first patch in Fig.3 (g). Then, the updated appearance representation is used to track the next coming frame. Our tracker repeats the processes from Fig.3(c) to Fig.3(g) to compute all the tracking results of a video.

In details, our contributions are made by the above three components of the proposed tracker. First, by our appearance model, the appearance representation of target supports the different variations of each patch, which performs enough flexible and rich cues to adjust the complex and unpredictable appearance changes of target and its surroundings. Second, the new basic tracker suppresses the noises and disturbs from the surroundings of tracking patch and boosts the response peak on the right target position, which finally leads to favorable tracked result. Third, the voting scheme greatly utilizes the structure information between patches and the whole target, which not only preserves the global structure of target but also accurately satisfies the quickly and drastically changes of its local patches by preserving the successful tracked patches and rejecting the failed tracked patches. The details about the proposed tracker are described as follows.

3.1 The appearance representation model of target via its multi-scale patches

For many changes of target appearances start and happen on its local regions, we represent the appearance of target by defining the local appearance representations of its multi-scale patches using the kernelized correlation filters. The proposed appearance representation model of target is formed by three steps. First, we randomly extract patches with different sizes around the given target region on the first frame. Figure 4 (a) shows an example of extracting multi-scale patches. The yellow rectangle describes the target region, while the orange rectangle is the extended target region and the red rectangles with dotted lines are the target patches. All the centers of patches must locate in the orange rectangle.

Second, we compute the relative distance d between the center of each patch and the center of target. In Fig. 4 (b), the Cp is the center of the green patch while the Ct is the center of target, and the d between them is their distance. At the same time, each patch is denoted by a mark b to describe it's a positive or negative one for the target detection. If the center of patch locates in the target region namely the yellow rectangle in Fig. 4 (a), its mark b is assigned to 1 to



Extracting multi-scale patches

Denote the center, mark, distance of patches

Fig. 4 To extract and denote the multi-scale patches of target object

show it is a positive patch, such as the red patch and the pink patch in Fig.4 (b). Otherwise, if the center of patch is in the extended region of target such as the region between the orange rectangle and the yellow rectangle, its mark b is assigned to -1 to show it is a negative patch, such as the green patch in Fig. 4 (b).

Third, we construct the initial appearance representation of each patch by the kernelized correlation filters proposed in [15]. This method uses the cyclic matrix to extract samples with same size around the center of a patch, and it utilizes these samples to train the classifier to represent the patch. The process of training is achieved by the ridge regression which successfully use the diagonalizing feature of cyclic matrix in the Fourier transformation to reduce the computing complexity. Using this method, each patch is described by a feature matrix y and a classifier matrix α. As proposed in [15], we use the Histogram of Oriented Gradient (HOG) in a high dimension space to construct the feature matrix. For each video, the feature matrix y on the first frame is the HOG of the given target region. However, in the following frames, it is obtained by the method defined in section 3.2 because the target region is unknown and need to be computed.

In addition, when a patch undergoes some great challenges such as complete occlusion, its tracking will be failed but simultaneously the tracking of other patches maybe is successful. Therefore, the times of being successful tracked for each patch is different. Here, we define another parameter named the successful times v to record the times that a patch is continuously and successfully tracked. In this paper, we define a tracking is successful when the intersection of two adjacent results is more than half of their union region. Following the above processes, the appearance representation of a patch includes five parameters, namely the relative distance d, the mark b, the feature matrix y, the classifier matrix α, and the successful times v. For each patch the parameters (d, b, y, α, v) of its appearance representation are initialized based on the given target region on the first frame and will be updated in the subsequently tracking.

3.2 The basic tracker based on Gaussian mixture model and Kernelized correlation filters

A new basic tracker is defined to track each patch of the target by introducing the gaussian mixture model into kernelized correlation filters (KCF). As demonstrated in the present work [15], the KCF produces rapid tracked results and perform favorable in dealing with simple and rigid deformation of target. However, it is not robust because the surroundings about a patch usually have similar appearances, and this makes the KCF to produce many peak values around the right target position of the patch. Unfortunately, the tracked result of a patch is decided by the peak value of the filters. Therefore, the KCF usually mistakes the position with a peak value but not the right position as the tracked result of a patch. A good way to reduce such disturbs is to use gaussian mixture model to suppress the unimportant peaks.

Therefore, we construct the new basic tracker via introducing the gaussian mixture model into the kernelized correlation filters to form an optimized classifier. A loss function and regular term are defined to achieve this optimization based on the mix gaussian representation and the peak values computed by the kernelized correlation filters. By minimizing this loss function, an optimized classifier is formed to help us to obtain the right tracked result. The details are as follows.

(1) The Kernelized Correlation Filters

The trackers based on the kernelized correlation filters achieve tracking by the following three steps. First, it forms a classifier according to the present tracked result. Then for candidate sampling around the last target region, it uses the trained classifier to evaluate the response value of each pixel to be the target position. Finally, the position with biggest response is taken as the target position, and the target size is assigned to the same values with the last ones. The classifier is updated according the new tracked result to adapt to the appearance change of target. In this paper, the candidate sample is extracted according to the result on the last frame of the tracking patch.

For the *i′th* patch on frame *t*, if we use $\alpha_{t,\ i}$ to denote the classifier computed by kernelized correlation filters, use $y_{t,\ i}$to denote the feature matrix of the appearance model, use $x_{t,\ i}$ to denote the feature matrix of candidate sample, use *k* to denote the process of kernelized correlation filters, the response matrix $R(x_{t,\ i})$ of describing the possibility of each pixel to be the target position is computed by:

$$R\left(x_{t,i}\right) = k\left(x_{t,i}^{T}, y_{t,i}\right)\odot\alpha_{t,i} \tag{1}$$

where $x_{t,i}^{T}$ is the transformed matrix of $x_{t,\ i}$, $\odot$ is the dot product of matrixes.

According to kernelized correlation filters, if a pixel owns the biggest response value, this pixel is the tracked center of the corresponding patch. However, the response matrix usually has some redundant peak values, and sometimes the peak with the biggest value is not the right tracked result. Therefore, if directly using the response values, the kernelized correlation filters usually cannot obtain the ideal target result and lead to tracking drift or failure. In this paper, we introduce the gaussian mixture model to optimize the classifier produced by kernelized correlation filters to construct a more robust tracker by reducing the disturbs introduced by the redundant peak values of the response matrix.

(2) The Proposed Basic Tracker based on Gaussian mixture model

As described above, the redundant peak values of response matrix lead the tracked result drifting away from the ideal target position. So, to reduce such peak values and further outstand the response value of the ideal target position, we construct an optimized classifier by combining the gaussian mixture model and kernelized correlation filters.

Figure 5 is an example of using the optimized classifier produced by our proposed basic tracker to get more accurate tracking result. In the left column of Fig. 5, the red rectangle is the target region of basketball man, and the blue rectangle is the tracking patch of this target while the red point is its center. Just using kernelized correlation filters, it employs the traditional gaussian model to compute the response matrix such as the top one in the middle column of Fig.5. However, this matrix has many peaks, and the tracked result of the blue rectangle located on the head of the basketball man as shown in the top one of the right column of Fig. 5. This result drifts the tracking patch from the body of basketball man to his head, and introduces obvious tracking errors. However, using the optimized classifier of our basic tracker improved by gaussian mixture model, the response matrix becomes more reasonable and accurate as shown in the bottom one of the middle column in Fig. 5. By this optimized response matrix, the tracked result of the blue rectangle located on the ideal target position as shown on the bottom of the right column in Fig. 5.

We design the proposed basic tracker to get the optimized classifier by the following four steps:

1) Getting some particles based on the peak values of response matrix. Using the Eq. 1, we compute the response matrix of the candidate sample of a tracking patch. Usually this matrix has many peaks as shown in the top image of the middle column in Fig. 5. We order these peaks from big to small and take the first p positions as the centers of particles. In our experiments, p is assigned to be 9. The size of the particles is the same with the tracking patch.
2) Using particles to achieve the representation of gaussian mixture model. After getting the particles for a patch, we use the classifier of kernelized correlation filters to compute the response matrix of each particle. Therefore, we compute p response matrixes for one patch. For each matrix of the patch, we extract its biggest response value. Then, using all the biggest responses of a patch as input, we construct the mixed gaussian representation for the patch. Figure 6 is an example of mixed gaussian representation. In the left image in Fig. 6, the red rectangle and the yellow rectangle are the target region and its extended region separately. We use 8 biggest response values from 9 particles to interpolate the

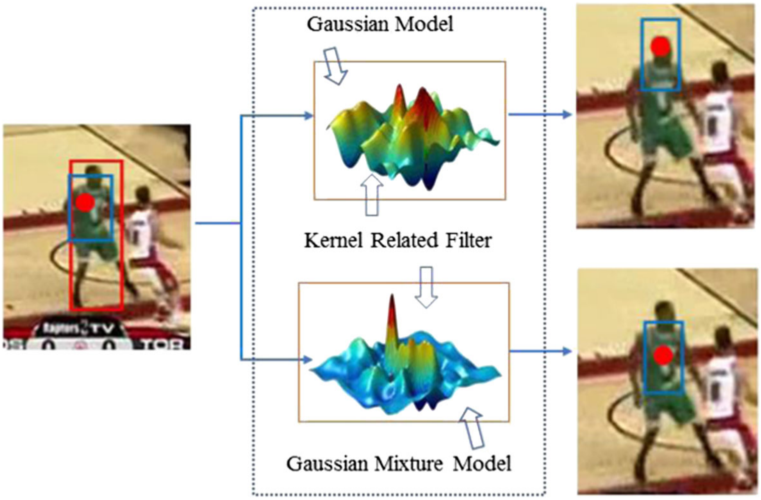


Fig. 5 The Tracking results of classifiers with Gaussian Mixture Model

gaussian mixture model formed by the biggest value of the 9 response matrixes. Then, we get the gaussian mixture model as the right image of Fig. 6.

3) Constructing a new loss function for our basic tracker. After obtaining the gaussian mixture model of a tracking patch, we introduce it as a regular term into kernelized correlation filters to define a new loss function. The loss function for the $i'th$ patch on frame t is defined as:

$$\min_{w_{t,i},y_{t,i}} \left\|\hat{x}_{t,i}w_{t,i}-y_{t,i}\right\|_2^2+\lambda_1\left\|w_{t,i}\right\|_2^2+\lambda_2\left\|y_{t,i}-y_{t,i_0}\right\|_2^2 \tag{2}$$

Where $\hat{x}_{t,i}$ is the corresponding matrix in the Fourier field of the feature matrix $x_{t,\ i}$ about the present candidate region of the tracking patch, $w_{t,\ i}$ is defined as the following Eq. 4 by $\hat{\alpha}_{t,i}$, and $\hat{\alpha}_{t,i}$ is the corresponding matrix in the Fourier field of the classifier matrix $\alpha_{t,\ i}$. The $y_{t,\ i}$ follows the noise model $y_{t,i}\sim N\left(y_{t,i_0}, diag^{-1}(1/(2\lambda_2))\right)$, and $y_{t,\ k}$ is the $k'th$ model of the gaussian mixture model:

$$y_{t,k}\sim N_k\left(\tilde{x}_{t,k}, u_k, \sigma_k^2\right) \tag{3}$$

Where $\tilde{x}_{t,k}$ is the samples of patch $x_{t,\ i}$, u_k and σ_k^2 are the expectation and variance of the $k'th$ gaussian model. All the gaussian models for patches form the gaussian mixture model of Eq. 2. We compute the matrix $y_{t,i_0}$by the gaussian interpolation and the correlations between the tracked results of patch $x_{t,\ i}$ from the last two frames. The first feature map of $y_{t,\ i}$ namely its value on the first frame is the HOG extracted from the user given target region. However, in tracking process, the target region of the tracking patch is unknown. Therefore, $y_{t,\ i}$ is computed by Eq.2 and Eq.3. We use the multiple template solution method proposed in [1] to solve the Eq.2.

4) Calculating the optimized classifier of our basic tracker. The optimized classifier of our basic tracker is defined based on the loss function. The computation of Eq. 2 belongs to the problem of ridge regression. As all the related parameters from the Fourier field, we need transfer the solution of Eq. 2 into the dual domain.

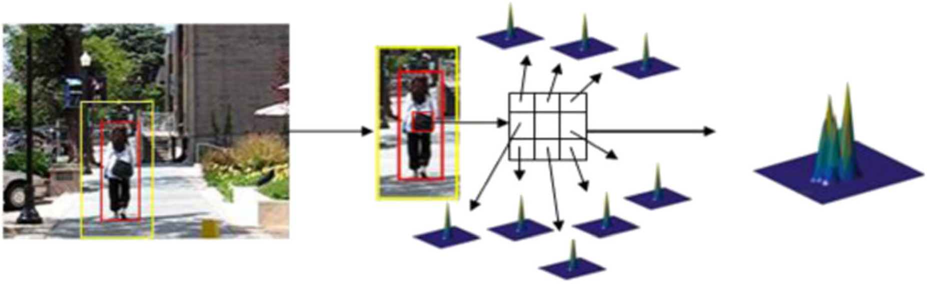


Fig. 6 To construct the Gaussian mixture model

The $w_{t,\ i}$ in the dual domain is computed by:

$$w_{t,i} = x_{t,i}^{T} \odot \hat{\alpha}_{t,i} \tag{4}$$

Therefore, by the Eq. 2 and Eq. 4, we compute the optimized classifier in the Fourier field $\hat{\alpha}_{t,i}$ by:

$$\hat{\alpha}_{t,i} = \frac{\left(\frac{\lambda_2}{\lambda_1}\left(\hat{x}_{t,i_1} \odot \hat{y}_{t,i_1}\right) + \lambda_2\right) \odot \hat{y}_{t,i_0}}{\frac{\lambda_2}{\lambda_1}\left(\hat{x}_{t,i_1} \odot \hat{y}_{t,i_1} \odot \hat{x}_{t,i_1} \odot \hat{y}_{t,i_1}\right) + \frac{1+2\lambda_2}{\lambda_1}\left(\hat{x}_{t,i_1} \odot \hat{y}_{t,i_1}\right) + (1+\lambda_2)} \tag{5}$$

where λ_1 and λ_2 are the parameters of the normal terms, and set to be (10^{-3}, 10^{-4}) in our experiments. $\hat{x}_{t,i_1}$ and $\hat{y}_{t,i_1}$ are separately the FFT of the first row of $x_{t,\ i}$ and $y_{t,\ i}$, where all the operations are element wise to reduce time cost. In this paper, we utilize the same method proposed in [1] to compute $\hat{x}_{t,i_1}$, $\hat{y}_{t,i_1}$ based on the $x_{t,\ i}$, $y_{t,\ i}$. Here, $x_{t,\ i}$ is the present candidate and the feature matrix $y_{t,\ i}$ is the appearance model of the tracking patch.

(3) The Tracked Result of a Patch

Using Eq. 2 and Eq. 4, we use Eq. 6 to train the new optimized classifier $\hat{\alpha}_{t,i}$. Then according to the theoretical basis of Eq. 1, we use the optimized classifier to construct the new response map $R'(x_{t,\ i})$ for patch $x_{t,\ i}$ by:

$$R'\left(x_{t,i}\right) = k\left(x_{t,i}^{T},\ y_{t,i}\right) \odot \hat{\alpha}_{t,i} \tag{6}$$

where $\hat{\alpha}_{t,i}$ is the optimized classifier, $y_{t,\ i}$ is the feature matrix of the appearance model for the $i'th$ patch on frame t, and $x_{t,i}^{T}$ is the transformed matrix of feature matrix of the candidate sample $x_{t,\ i}$. The position with biggest value in $R'(x_{t,\ i})$ is the center of the tracked result $pos_{t,\ i}$ of the $i'th$ patch on frame t.

3.3 Computing the final tracked result of the whole target

Our proposed tracker computes the target region based on the tracked results of all its patches. Three factors decide the target region of a tracking object. One is the probability value of each patch which describes the probability that the patch's result belonging to the target patch. Another one is how to deal with the failed tracked patches which can introduce tracking drift or failure if they are used in a wrong way. The third one is how to use the Hough vote to compute the final tracked result of the whole target when the above two factors have been solved.

(1) The Probability Value of a Patch

Having the tracked result of each patch, we calculate the probability value of its result belonging to target patch by:

$$p\left(x_{t,i}|y_{t,i}\right) = p_t\left(x_{t,i}|y_{t,i}\right)p_0\left(x_{t,i}|y_{t,i}\right) \tag{7}$$

where $x_{t,\ i}$, $y_{t,\ i}$ have the same meanings as above, p_t, p_0 are the patch confidence and patch target probability.

The patch confidence p_t is based on the response value of its tracked result. As proposed in [3], we still use the Peak-to-Sidelobe Ratio (PSR) to compute the patch confidence. For patch $x_{t,\ i}$, its patch confidence is defined by:

$$p_t\left(x_{t,i}|y_{t,i}\right)=\left(\frac{max\left(R^{'}\left(x_{t,i}\right)\right)-\mu_{\Phi}\left(R^{'}\left(x_{t,i}\right)\right)}{\sigma_{\Phi}\left(R^{'}\left(x_{t,i}\right)\right)}\right)^2 \tag{8}$$

where $R'(x_{t,\ i})$ is the response matrix computed by Eq. 6, Φ is the surrounding region of the biggest value in $R'(x_{t,\ i})$, μ_{Φ}, σ_{Φ} are the mean value and the standard deviation of the response values in $R'(x_{t,\ i})$ out of region Φ.

The patch target probability p_0 describes the stability and continuity of a patch being successfully tracked. If a patch can be continuously and successfully tracked as a positive patch, it plays more important role in deciding the final target, and should have a big probability value. In this paper, we use $l(x_{t,\ i})$ to describe the situation that a patch being continuously tracked as a positive patch. Then, we define patch target probability by:

$$p_o\left(x_{t,i}|y_{t,i}\right)=e^{l\left(x_{t,i}\right)} \tag{9}$$

where $l(x_{t,\ i})$ is defined based on the numbers of patch $x_{t,\ i}$ being continually and successfully tracked, and its positive or negative mark. We calculate $l(x_{t,\ i})$ by:

$$l\left(x_{t,i}\right)=b_{t,i}\left(\frac{1}{n^-}\sum_{j\in\Omega^-}\left\|v-v^{(j)}\right\|_2-\frac{1}{n^+}\sum_{i\in\Omega^+}\left\|v-v^{(i)}\right\|_2\right) \tag{10}$$

where $b_{t,\ i} \in \{+1, -1\}$ is the positive or negative mark of patch $x_{t,\ i}$. Ω^+ is the set of patches with positive mark, namely their marks are equal to 1, and n^+ is its number of patches. Ω^- is the set of patches with negative mark, namely their marks are equal to −1, and n^- is its number of patches. The v is the times of patch $x_{t,\ i}$ being continuously and successfully tracked.

(2) Dealing with the failed tracked patches

For the tracking challenges emerge randomly and unpredictably, the tracking failure or drift of patches usually cannot be avoided. If the failed tracked patches are used in compute the final tracked result of the whole target, they will bring great tracking errors. Therefore, we propose to resample the failed tracked patches to avoid such errors.

In this paper, we search the failed tracked patches according to the following three steps. First, we select the patch whose tracked center is not in the extended region of the last target result. Because such patches cannot represent the local region of target or the its surrounding background. We delete this kind of patches. Second, we compute the ratio between the positive tracked patches and the negative tracked patches. If the ratio is too big, we delete some positive patches with smaller response values. If the ratio is too small, we delete some negative patches with smaller response values. The smaller response value means the less probability that it being successfully tracked. Third, we delete the patches whose response values are too low, because the tracked results of these patches usually have great errors.

After deleting the failed tracked patches, we resample some new patches to make a supplementary. We first select the tracked center of the patch owning the biggest probability value based on Eq. 6. Then, we resample new patches around this center. The number of resampled patches is the same with the deleted ones. Finally, we use the Eq. 1 to build the appearance representations of the new patches and use Eq. 7 to compute these probability values.

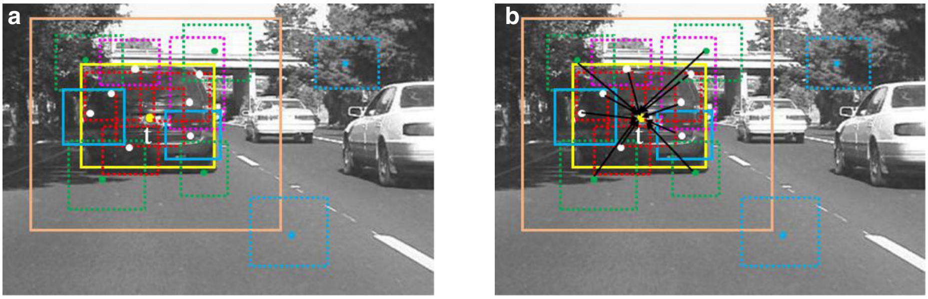
(3) Computing the final tracked result of the whole target

With the probability values of the reserved and resampled patches, we compute the final tracked result of the whole target by Hough vote [9]. Figure 7 demonstrates an example of the vote from patch centers to target center. In Fig. 7, the yellow point is the target center, while the white, green and blue points are its patch centers. The yellow rectangle describes the target region from last frame, and the orange rectangle is the extended region of this region. The green, red, pink and blue rectangles are the patches with different sizes. As shown in Fig. 7, the two centers in the blue dotted rectangles locate outside the extended region, they are failed tracked and rejected to do vote. But we resample two new patches as the blue solid rectangles to supplement them. In addition, as the centers of red and pink patches locate in target region, they are the positive and successful patches. Similarly, as the centers of the green patches locate outside the target region but inside the extended region, they are the negative and successful patches. All the successful patches and resampled patches take part in the vote of the final tracked result. Figure 7 (b) uses the black lines with narrows to demonstrate the support vectors of Hough vote from patch centers to target center.

According to the Hough vote process, the target center is decided by three factors, namely the patch centers, the relative distance between patch center and target center, the probability value of each patch. In details, the target center on frame t is computed by:

$$Tpos(t) = \sum_{i=1}^{n} \left(pos_{t,i} + d_{t,i}\right) \times p\left(x_{t,i} | y_{t,i}\right) \tag{11}$$

where $Tpos(t)$ is the target center on frame t, $p(x_{t,\ i}| y_{t,\ i})$ is computed by Eq. 7 and denotes the probability value of patch $x_{t,\ i}$,$pos_{t,\ i}$ is the patch center of $x_{t,\ i}$ decided by biggest response value from Eq. 6. $d_{t,\ i}$ is the relative distance between the patch center of the $i'th$ patch and target center, but it is computed based on the results on the last frame.



the centers of target and patches

the vote from patches to target

Fig. 7 The Hough vote from patch centers to target center. The yellow point is the target center, while the white, green and blue points are the patch center. The yellow rectangle is target region while the orange one is its extended region

3.4 Appearance representation model update

To adapt to the unpredictable and complex changes of target object and its surroundings, we separately update the appearance representations of all patches according to their tracked results. This update is classified into two kinds. One is to update the failed tracked patches, and the other is to update the successful tracked patches.

(1) Updating the failed tracked patches

For the failed tracked patches, we first delete them and then resample new patches with equal number by the approach proposed in section 3.3 to supplement them. Finally, we use kernelized correlation filters to build the appearance representations of the new resampled patches. When we go on tracking target in subsequence images, these appearance representations of new patches are employed as the local representations of target, while the failed tracked patches and their appearance representations are discarded.

(2) Updating the successful tracked patches

For the successful tracked patches, we define the algorithm to compute its updated appearance representations via two steps. First, we use the kernelized correlation filters and the new tracked result of each patch to compute its new feature matrix and its new classifier matrix. These two new matrixes describe the new appearance information coming from the recent result of the patch. Second, we combine the new appearance representation with the old appearance representation to get the updated appearance representation of each patch. Equation 12 describes the process about this kind of update.

$$\begin{cases} y_{t+1,i} = \omega y_{t,i} + (1-\omega) y^{'}_{t+1,i} \\ \alpha_{t+1,i} = \omega \alpha_{t,i} + (1-\omega) \alpha^{'}_{t+1,i} \end{cases} \tag{12}$$

where $y_{t+1,i}$ and $\alpha_{t+1,i}$ are the updated feature matrix and updated classifier matrix to represent the appearance of the $i^{'}th$ patch on frame $t+1$, $y_{t,i}$ and $\alpha_{t,i}$ are the feature matrix and classifier matrix of the $i^{'}th$ patch on frame t, $y^{'}_{t,i}$ and $\alpha^{'}_{t,i}$ are the new feature matrix and the new classifier matrix computed based on the new tracked result on frame t of the $i^{'}th$ patch. The ω is a constant coefficient and we set $\omega = 0.3$ in our experiments.

As described in Section 3.1, the appearance representation of each patch has six parameters. Therefore, after updating the feature matrix and classifier matrix, we need to update the other four parameters of the appearance representation for each patch, including the relative distance, the negative or positive mark and the successful times. These parameters are computed based on the new tracked results of each patch and tracking target.

4 Experiments and evaluations

The proposed tracker was implemented 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. We use the TB-50 database proposed in the Tracking Benchmark [42] to verify the proposed tracker. 8 trackers are used to do comparisons, including the recent famous trackers

DLSSVM (CVPR'16) [34], KCF(PAMI' 15) [15], RPT (CVPR' 15) [22], LSHT(CVPR '13) [13], LSST(CVPR' 13) [11] and the top three ranked trackers from the Tracking Benchmark namely the STRUCK (PAMI'16) [12], ALSA (CVPR'12) [19], SCM (CVPR'12) [47]. Table 1, Table 2, Fig.7 and Fig.8 describe the quantitative evaluation, while Fig.9 shows the qualitative evaluation. More comparisons about all the videos and challenges are attached in the supplement files. The evaluations and comparisons demonstrate that the proposed tracker produces more accurate result and performs much more favorable in dealing with coexisting challenges such as occlusion, deformation and so on.

4.1 Database and trackers

The TB-50 data used in our experiments include 50 videos and including many different tracking challenges. Each video has more than one challenges, and the situation that some challenges emerge at the same time is very common in these videos. All the tracking challenges can be divided into 11 classifications according to the method proposed in [42], including object deformation (DER), occlusion (OCC), illumination variation (IV), fast moving (FM), background clutter (BC), in-plane rotation (IPR), low resolution (LR), out-plane rotation (OPR), out-of-view(OV), moving blurs (MB), scale variation (SV). In the tracking benchmark [42], there is a TB-100 data which includes 100 videos to evaluate trackers. However, we just use the 50 videos in the TB-100, namely the TB-50 to do evaluation. The main reason is that as described in [42] these 50 videos have been repeatedly verified its effectiveness in evaluating trackers and most of the present trackers supplied the tracked results of these 50 videos for comparison such as the LSST, ALSA and so on. For each video in TB-50, the tracking object is specified on the first frame by the people who firstly provide the video. In our evaluation, we use the David (300:770) and Freeman4(1:283) to do comparison. For the other 48 videos, we employ all the frames of the videos to achieve the tracking and comparing.

As popular in the recent work [35, 37, 42], we choose the precision plot and the success plot to do the quantitative evaluations of our tracker. The success plot describes the percentage of successfully tracked frames which is decided by the Intersection Over Union (*IOU*). Bigger success plot means better results. The *IOU* is computed by $(R_T \cap R_G)/(R_T \cup R_G)$ where ∩ and ∪ are the intersection and union of tracked box (R_T) and ground truth (R_G), respectively. In our experiments, when the *IOU* of a frame is bigger than 0.5, we denote it is successfully tracked. The precision plot is defined by the percentage of successfully tracked frames based on the Center Location Error (*CLE*) with a given threshold T_C ($T_C = 20$ in our experiments). Bigger

Table 1 The success plot of the 9 trackers on the TB-50 with 11 challenges

Trackers	ALL	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
OURS	0.435	0.411	0.462	0.401	0.398	0.442	0.430	0.340	0.384	0.420	0.359	0.404
KCF	0.400	0.361	0.388	0.380	0.415	0.398	0.397	0.313	0.375	0.380	0.294	0.350
RPT	0.437	0.428	0.462	0.423	0.406	0.463	0.413	0.350	0.388	0.420	0.368	0.406
DLSSVM	0.424	0.400	0.428	0.466	0.387	0.405	0.438	0.384	0.420	0.407	0.465	0.379
LSHT	0.338	0.362	0.374	0.393	0.310	0.312	0.341	0.327	0.339	0.347	0.381	0.310
LSST	0.280	0.247	0.244	0.239	0.219	0.242	0.259	0.254	0.287	0.255	0.274	0.280
STRUCK	0.384	0.409	0.362	0.406	0.323	0.330	0.381	0.319	0.336	0.335	0.340	0.362
SCM	0.371	–	0.385	–	0.322	0.416	0.340	0.461	0.374	0.365	0.316	0.383
ASLA	0.334	–	0.347	–	0.291	0.365	0.315	0.440	0.341	0.328	–	0.344

Table 2 The precision plot of the 9 trackers on the TB-50 with 11 challenges

Trackers	ALL	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
OURS	0.592	0.537	0.581	0.557	0.534	0.604	0.589	0.537	0.539	0.564	0.460	0.568
KCF	0.556	0.483	0.487	0.501	0.558	0.545	0.534	0.510	0.527	0.527	0.374	0.511
RPT	0.590	0.555	0.571	0.572	0.525	0.631	0.558	0.549	0.529	0.553	0.441	0.567
DLSSVM	0.574	0.516	0.541	0.597	0.538	0.546	0.585	0.589	0.579	0.569	0.612	0.533
LSHT	0.444	0.452	0.458	0.480	0.395	0.389	0.445	0.508	0.466	0.474	0.517	0.419
LSST	0.379	0.304	0.333	0.290	0.315	0.332	0.346	0.400	0.405	0.361	0.374	0.374
STRUCK	0.384	0.409	0.363	0.406	0.323	0.330	0.381	0.319	0.336	0.335	0.340	0.362
SCM	0.371	–	0.385	–	0.322	0.416	0.340	0.462	0.374	–	0.316	0.383
ASLA	0.334	–	0.347	–	0.291	0.381	0.315	0.441	0.341	0.328	–	0.344

precision plot means better tracking results. For more details about *CLE*, *IOU*, please review the work in [42].

4.2 Quantitative evaluation

The average values of the precision plot and success plot about TB-50 are the important factors to evaluate the effectiveness of a tracker. Table 1 describes the average success plot about the 50 videos in TB-50 of our tracker and the present and famous 8 trackers. The red, green and blue numbers separately describe the best, second and third tracker. As shown in Table 1, our tracker always performs as the top three in all tracking challenges. Especially, when dealing with the background clutter (BC) and out-plane rotation (OPR), it performs as the best tracker among the 9 trackers. It demonstrates that our method can accurately track the whole target by combing the gaussian mixture model and kernelized correlation filters.

Table 2 describes the precision plot about the 50 videos in TB-50 of our tracker and the present and famous 8 trackers. As Table 1, the red, green and blue numbers separately describe the best, second and third tracker. According to the values, we conclude that for all the 11 tracking challenges, our tracker performs as the first top three one. The average precision of all total 50 videos is 0.592, which is biggest one among all 9 trackers. It means that our tracker can robustly produces more favorable results on all the test videos than the 8 trackers. In addition, our tracker is ranked as the first one in dealing with the background clutter (BC), in-plane rotation (IPR) and scale variation (SV). It demonstrates that our proposed appearance

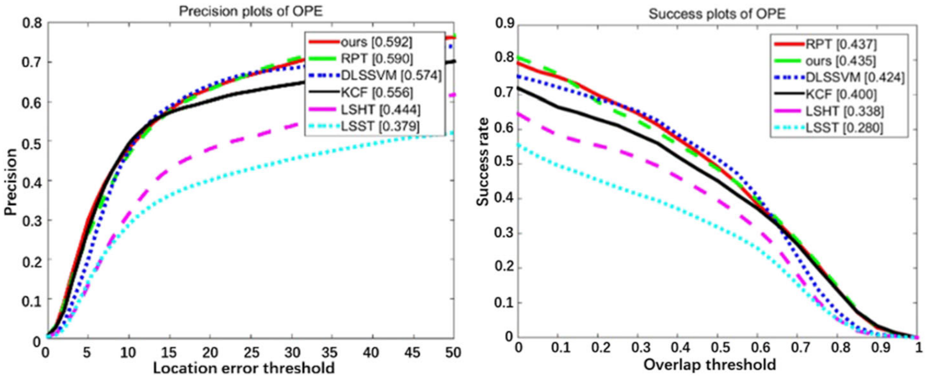


Fig. 8 The overall precision plot and success plot on TB-50 from the Tracking Benchmark [42]

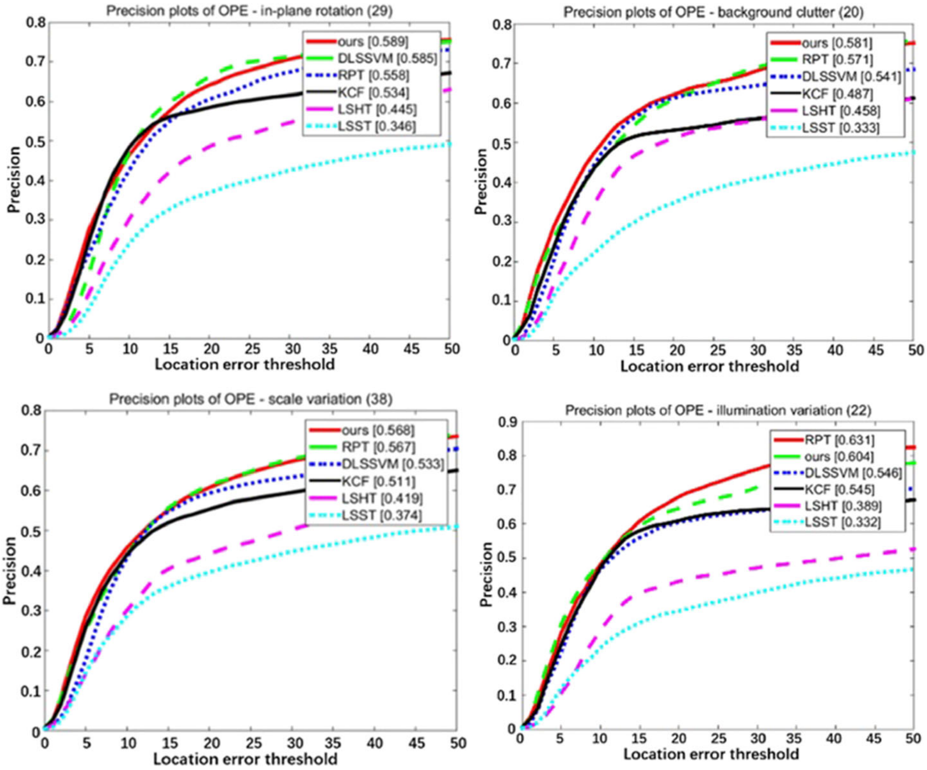


Fig. 9 The comparison of precision plot on the four kinds of tracking challenges, including the in-plane rotation, background clutter, scale variation and illumination variation

representation model based on multi-scale patches is very effective and efficient in adapting to the complex or drastic target changes.

Figure 8 shows the variation of the overall precision plot (the left one) and success plot (the right one) on TB-50 with different threshold. It is clearly that the bigger threshold leads to bigger precision plot or success plot. We select 6 trackers (including RPT, DLSSVM, KCF, LSHT, LSST) to do comparison, and they perform better in the 8 trackers and proposed in recent years. Compared with the 6 trackers, the precision plot of our results has the biggest precision plot value which demonstrates that according to this evaluation our method is the best tracker. Meanwhile, the success plot of our method is ranked as the second one. As shown in Fig. 8 (b), our success plot is 0.435 while the one of the best tracker (RPT) is 0.437, which means that our method only has a little lower value (0.2%) than the best one. The values and ranks in Fig. 8 successfully demonstrate that our method produces more accurate and robust results based on the precision and success on the total 50 videos of TB-50 provided in the Tracking Benchmark [42].

Figure 9 shows the precision plot about four kinds of tracking challenges, including the illumination variation (IV), background clutter (BC), in-plane rotation (IPR) and scale variation (SV). The precision plot about the other seven kinds are demonstrated in the attached file. We still use the 6 trackers (including RPT, DLSSVM, KCF, LSHT, LSST) to do comparison for their better performances. As shown in Fig. 9, our method has the biggest precision value in dealing with IPR (in-plane rotation), BC (background clutter) and SV (scale variation), which demonstrates that it is the best tracker in dealing with these challenges. For the challenge of IV (illumination variation), our method performs as the second tracker which means that it can gradually adapt to the change of target size. These evaluations demonstrate that our tracker can successfully transform the tracking challenges from the whole target onto its local patches. Therefore, our tracker greatly improves the precision plot and success plot by using the accurate tracked results of patches and the structure information between them.

4.3 Qualitative evaluation

Figure 10 demonstrates the tracked results of 6 trackers by the frame-to-frame comparison. The first and second rows of Fig. 10 show that our tracker produces more favorable results than the other 5 trackers in dealing with the scale variation and target shaking. The main reason is we uses kernelized correlation filters to track the multi-scale patches of target, which successfully adapt to the target scaling and the moving blurs. The third row of Fig. 10 shows that when the target is disturbed by the similar object, our tracker has not introduced tracking drift or failure. This good character is benefitted from the gaussian mixture model, because when it is introduced in kernelized correlation filters it can effectively reduce the disturbs from the adjacent surroundings and finally improve the tracking accuracy.

The fourth row in Fig.10 shows that our tracker successfully tracking the target in background cluster. That is because we update the appearance representation of target by separately updating the appearance representation of its patches. This process greatly preserves the appearances of patches with no or little changes, and quickly update the appearances of patches with drastic changes. Following this, our tracker not only successful adapts to the appearance change of target but also reduce the disturbs from background. This update scheme also makes our method can flexible and effectively adapt to the illumination variation as shown in the fifth row of Fig. 10. The sixth and seventh rows of Fig. 10 show that our tracker successfully deal with the great change or rotation of target. The reason is we detect target by tracking its patches, which leads the deformation of target shared by all the patches. The eighth row of Fig. 10 shows that our tracker produces ideal results in occlusion and disocclusion. That is because we track the patches with heavy occlusion by rejecting them and resampling new ones to supplement them. In a word, all these experiments in Fig. 10 demonstrate that our tracker is more reasonable and effective than many present trackers by introducing the gaussian mixture model into kernelized correlation filters, and by defining the scheme to track target based on identifying its multi-scale patches.

4.4 Implement efficiency

The implementing time of trackers is an important factor to evaluate them. Usually, people use the frames processed per second (fps) to describe the implement efficiency. However, for many trackers, its accuracy and efficiency are often restricted with each other. For example, to improve accuracy, their time cost will increase, while to reduce accuracy the time cost will decrease. In fact, people usually use the $IOU > 0.5$ and the $CLE < 20$ to denote the successful tracked result. They measure the implementing time while obtaining the maximus tracked results. Table 3 describes the fps of some most related trackers. The KCF produces online tracked results because it uses cyclic matrix to predict target. Our method and the RPT use KCF as the basic operator to track many local patches to vote the final tracked result.

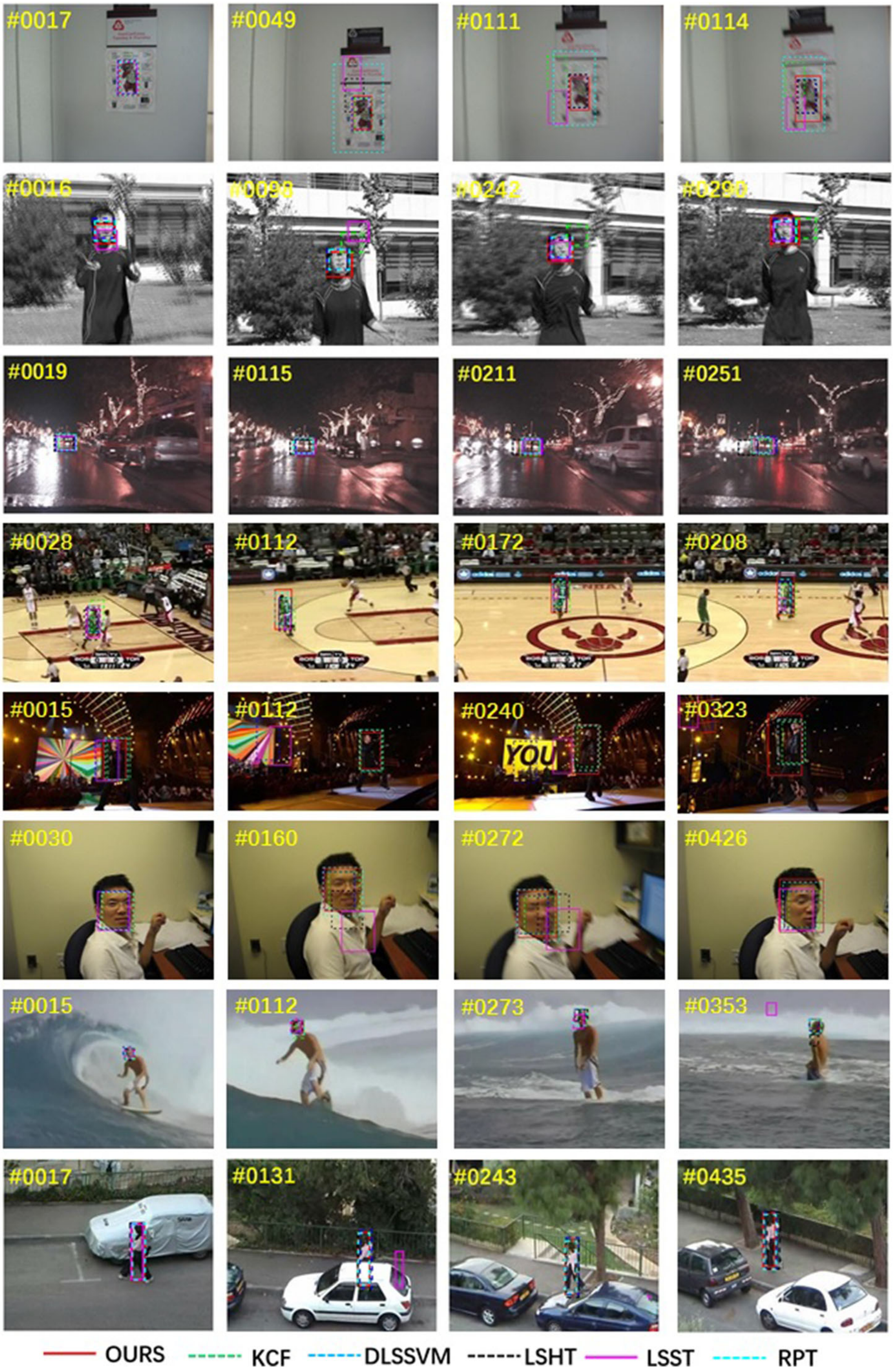


Fig. 10 The comparisons of different trackers with good performances on 8 videos

Table 3 The amount of frames processed per second (fps) with different trackers

Tracker	LSHT	ASLA	DLSSVM	STRUCK	SCM	KCF	RPT	OURS
FPS	25.6	1.5	25	10	0.42	321	4.5	6

Therefore, both the RPT and our method produce 4–6 frames per seconds. Our method is a little faster than RPT. The reason is that we reduce the number of patches while using the gaussian mixture model to improve the tracking accuracy of each patch. In addition, our tracker is robust and convergent in long-term tracking, because it overcomes the challenges which can lead to no convergence by separately and accurately deal with the challenges from patches.

5 Conclusion

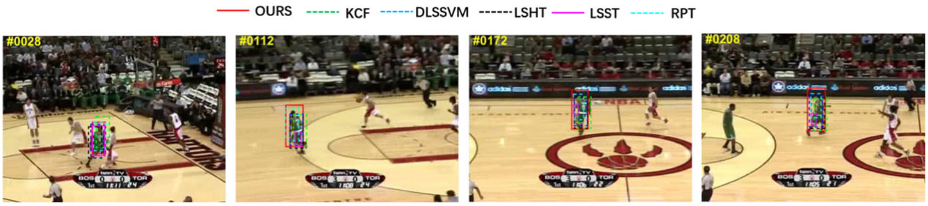
Visual tracking is a hot topic in computer vision, it is widely used in automatic driving, virtual reality, augmented reality, video surveillance, robotics and so on. However, the unpredictable and complex tracking challenges often emerge at the same time which usually lead to tracking failure or drift. In this paper, we propose a new robust tracker by identifying and tracking the multi-scale patches of target. This method defines the tracking scheme by combing the gaussian mixture model and the kernelized correlation filters, and it constructs a loss function with a normal term to track target patches. With the new loss function, the proposed tracker accurately tracks each patch by getting a response matrix with one outstanding peak. In addition, it uses the Hough vote to detect target based on its patches, which successfully preserve and inherit the structure constraints between them to gradually adapt to the photometric and geometric variations of target. Many quantitative and qualitative evaluations on TB-50 of the Tracking Benchmark have demonstrated that the proposed tracker has bigger success rate and less center locations errors. Therefore, compared with some famous present trackers, our tracker produces much more robust and accurate results in dealing with the coexisting tracking challenges.

However when tracking a frame, the proposed tracker implements our basic tracker for many times, because each patch requires a process. For example, if we extract 30 patches to represent the target, the proposed tracker will use 30 basic tracker to track these patches. Although the basic tracker is based on the high-speed KCF, so many processes leads to great time cost. Therefore, the proposed tracker cannot achieve the on-line tracking. In the future, we plan to introduce many constraints to reduce the number of patches while preserving the tracking accuracy.

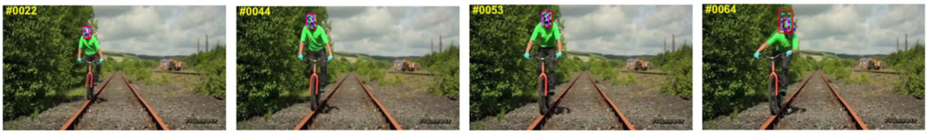
Acknowledgements We are grateful to all the reviewers for their valuable suggestions. We also greatly appreciate Professor Meihua Wang for her useful discussion and many helps in improving the manuscript. This work was supported in part by the National Natural Science Fund of China (61772209, 61472335) and the Science and Technology Planning Project of Guangdong Province (2016A050502050, 2014A050503057), the National Key R&D Program of China (2017YFB0503500) and the Zhejiang Provincial Natural Science Foundation of China (LY17F020009).

Appendix 1: The experiments on 50 videos of TB50 from Tracking Benchmark by frame-to-frame comparison

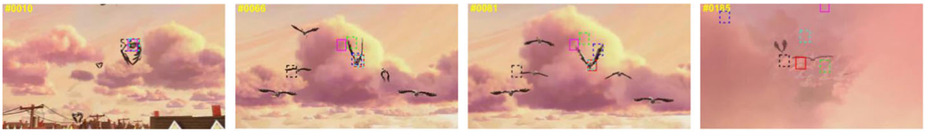
8 trackers are used to do comparisons, including the recent famous trackers DLSSVM (CVPR'16) [29], KCF(PAMI' 15) [27], RPT (CVPR' 15) [32], LSHT(CVPR '13) [39], LSST(CVPR' 13) [5] and the top three ranked trackers from Tracking Benchmark [13] namely the STRUCK (PAMI'16) [9], ALSA (CVPR'12) [3], SCM (CVPR'12) [12]. We select 6 trackers (RPT, DLSSVM, KCF, LSHT, LSST) to demonstrate the results, and they perform better in the compared 8 trackers and proposed in recent years. The references of these trackers are described at the end of this file.



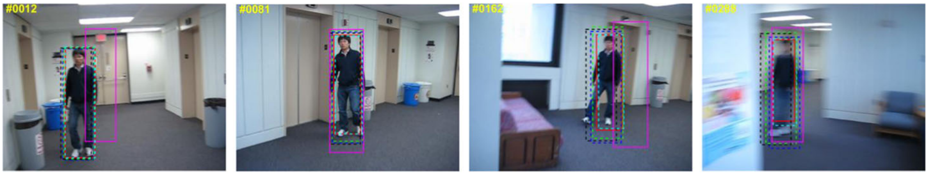
Video 1: BasketBall



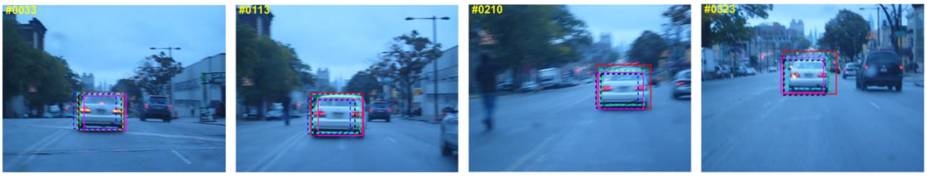
Video 2: Biker



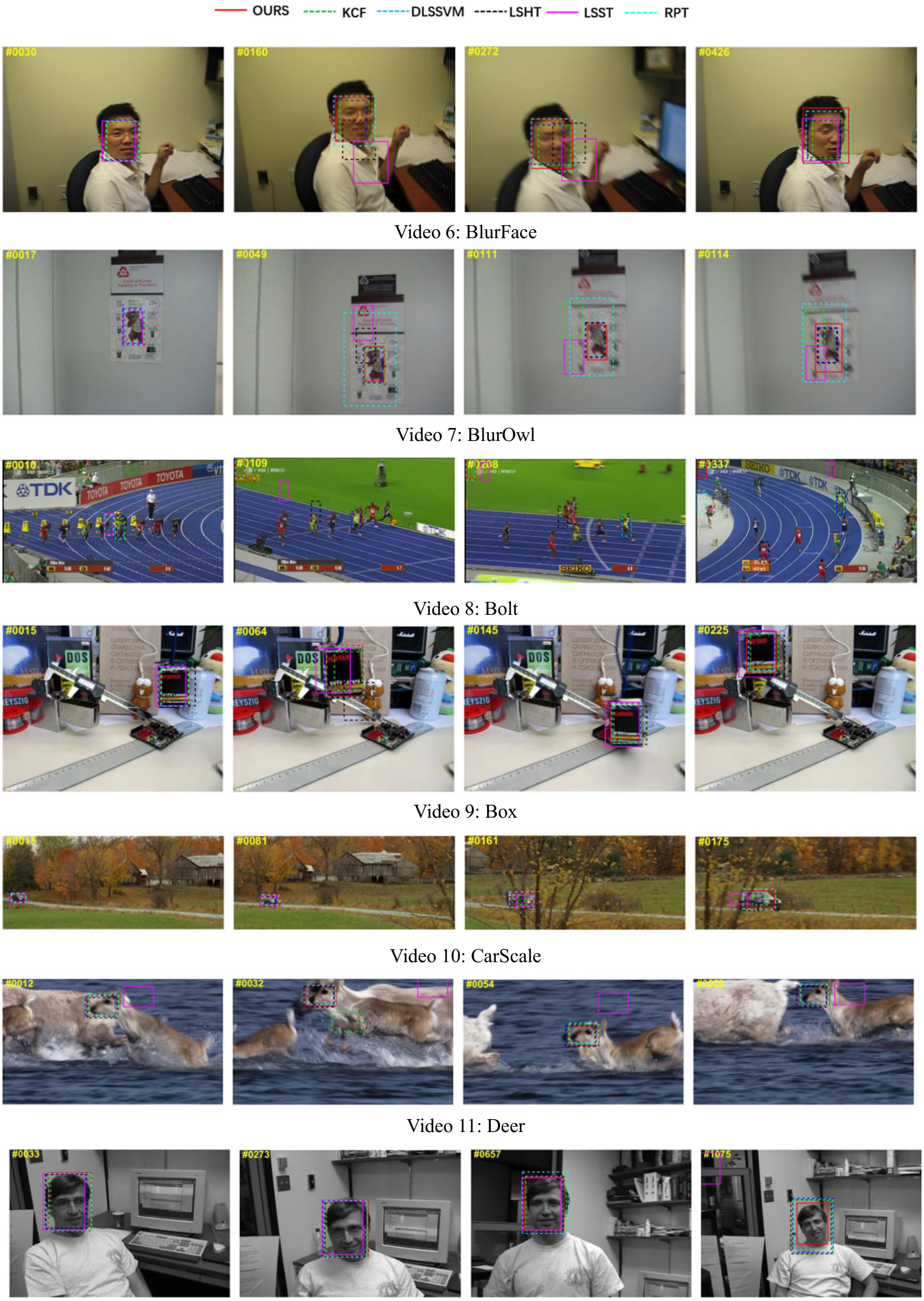
Video 3: Bird1



Video 4: BlurBody



Video 5: BlurCar2



Video 6: BlurFace

Video 7: BlurOwl

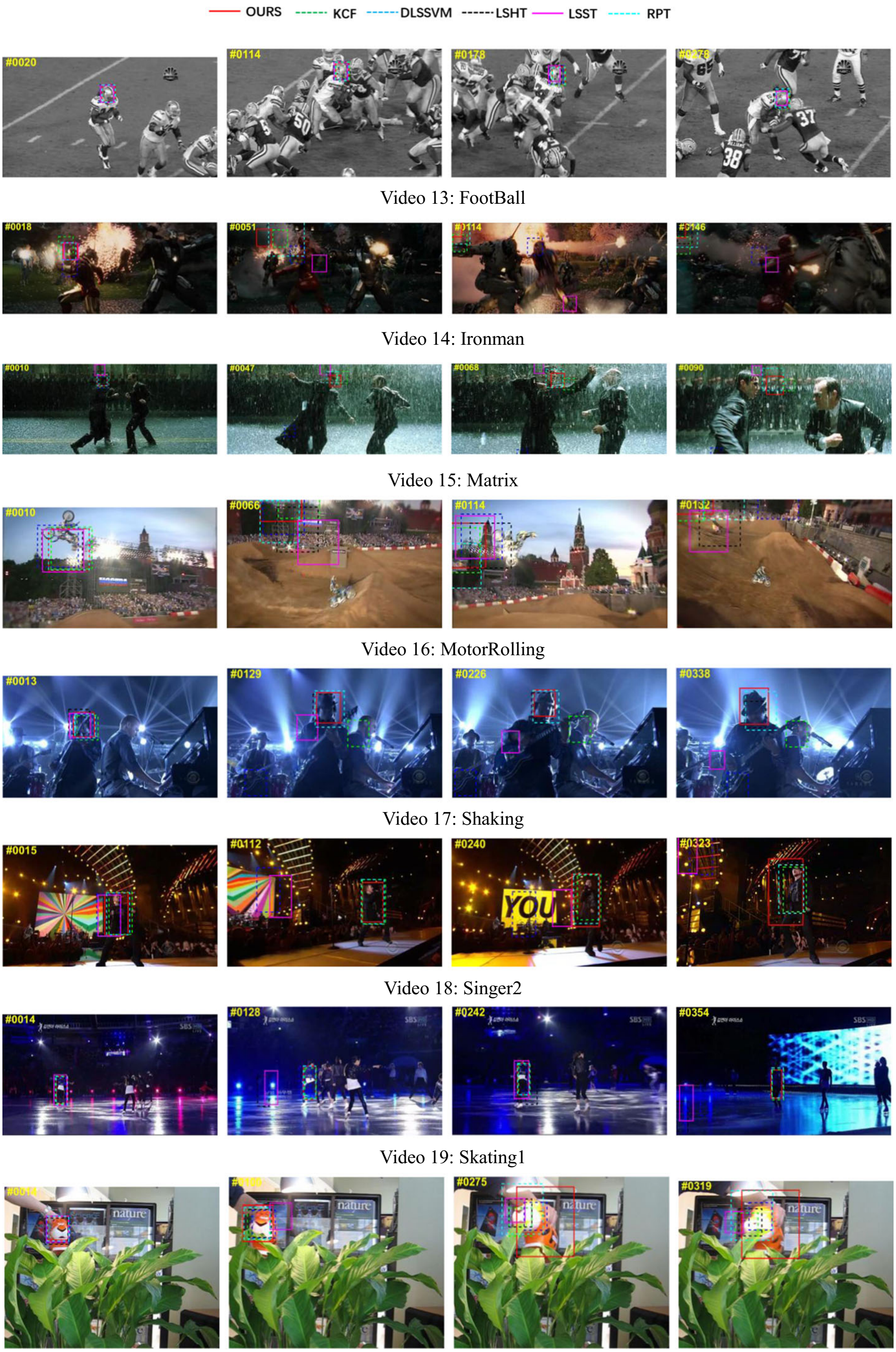
Video 8: Bolt

Video 9: Box

Video 10: CarScale

Video 11: Deer

Video 12: Dudek



Video 13: FootBall

Video 14: Ironman

Video 15: Matrix

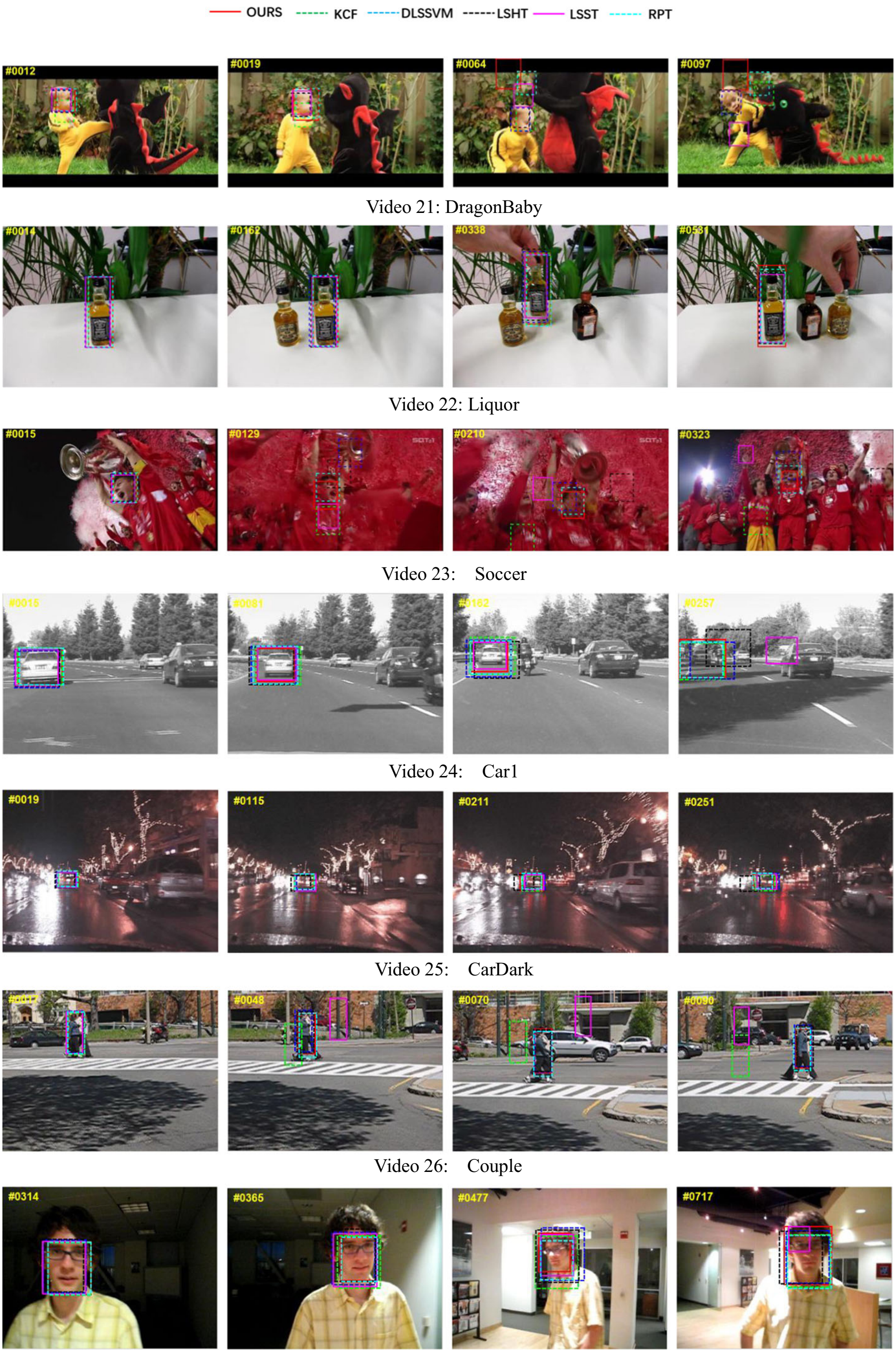
Video 16: MotorRolling

Video 17: Shaking

Video 18: Singer2

Video 19: Skating1

Video 20: Tiger2



Video 21: DragonBaby

Video 22: Liquor

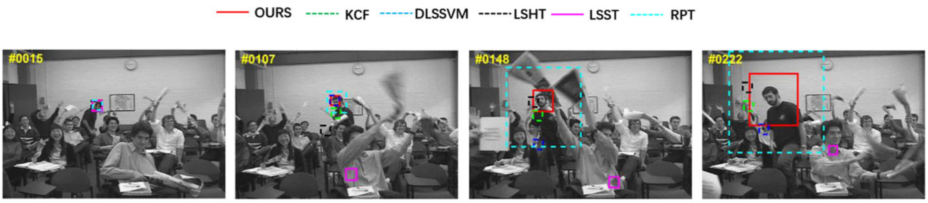
Video 23: Soccer

Video 24: Car1

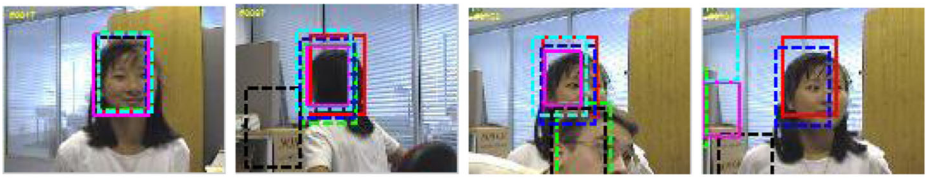
Video 25: CarDark

Video 26: Couple

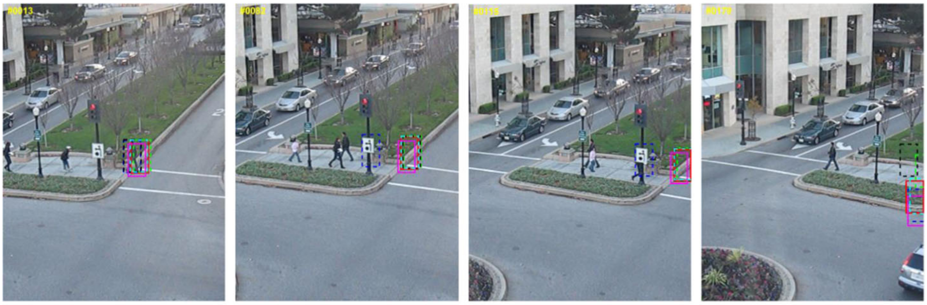
Video 27: David1



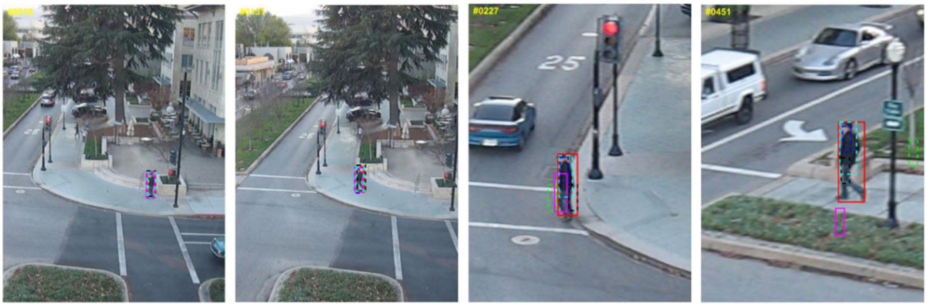
Video 28: Freeman4



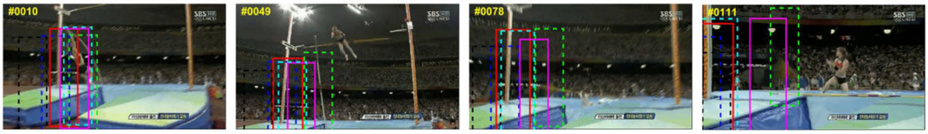
Video 29: Girl



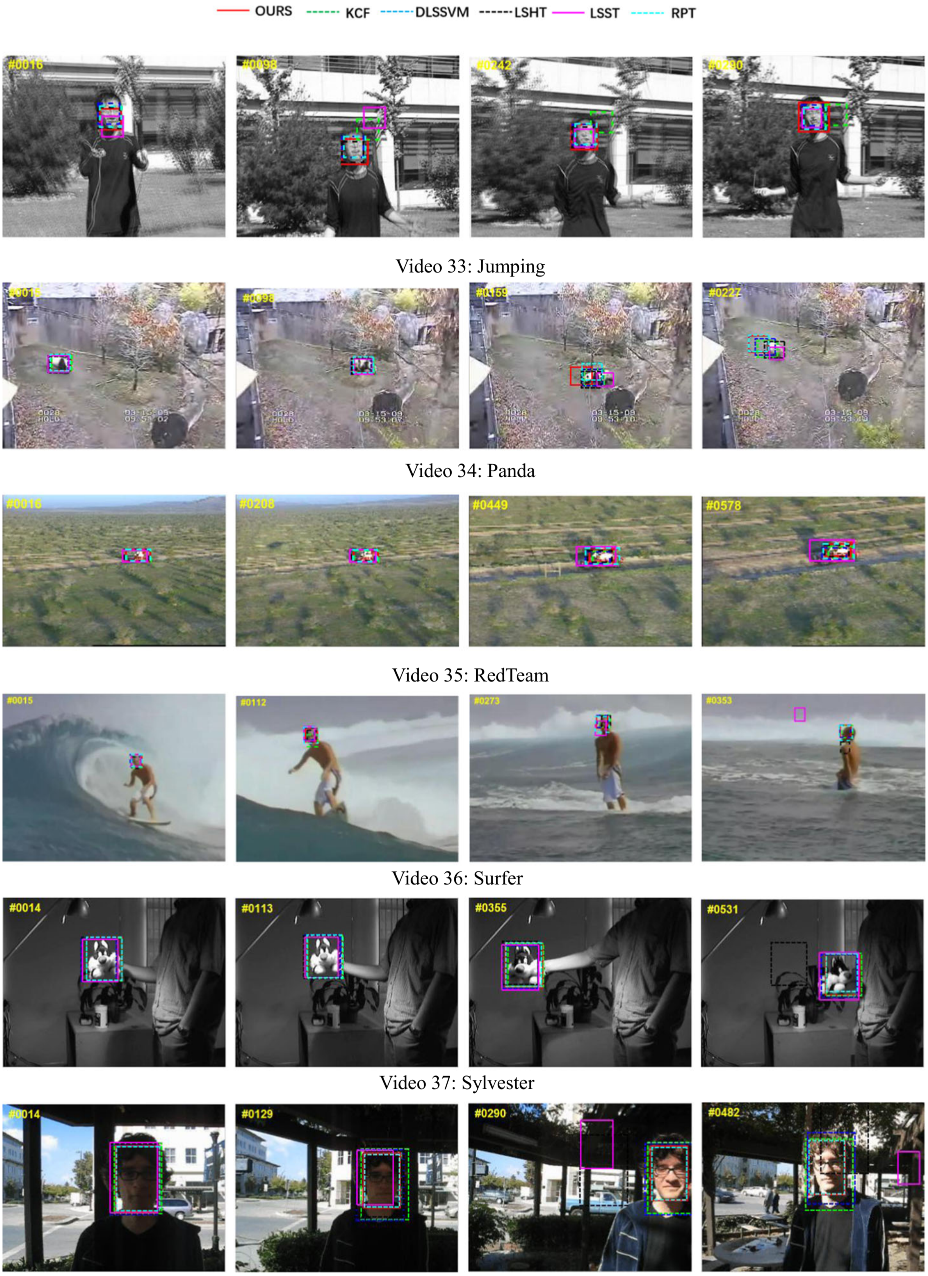
Video 30: Human3



Video 31: Human6



Video 32: Jump



Video 33: Jumping

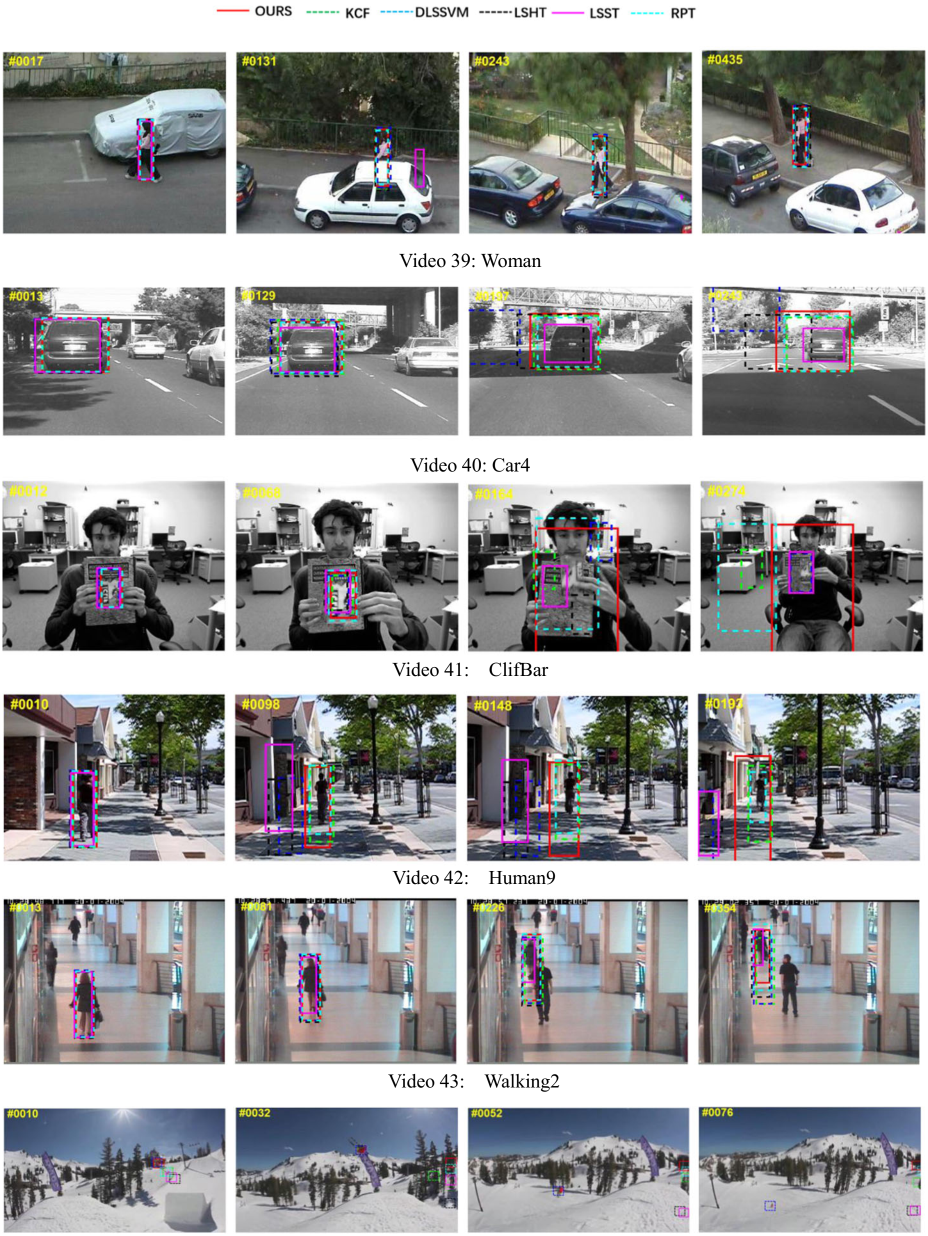
Video 34: Panda

Video 35: RedTeam

Video 36: Surfer

Video 37: Sylvester

Video 38: Trellis



Video 39: Woman

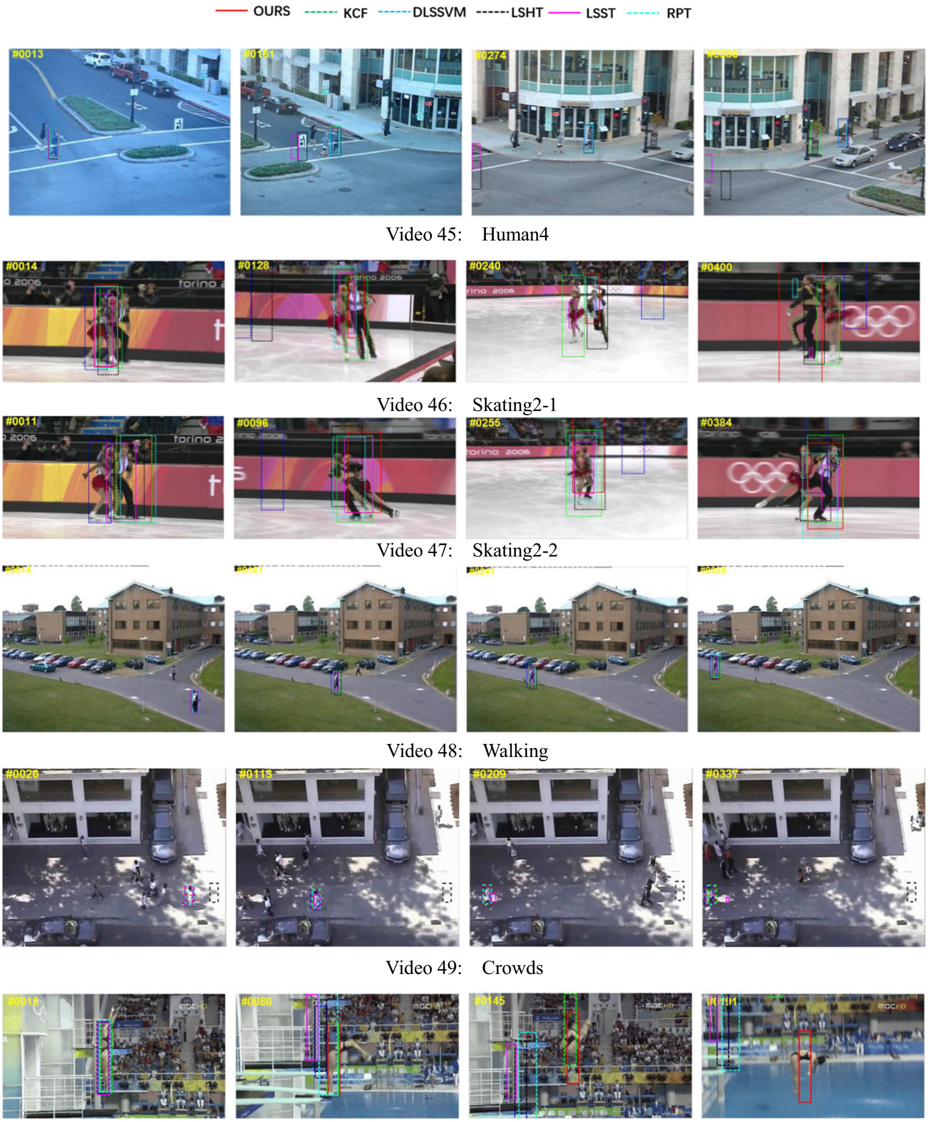
Video 40: Car4

Video 41: ClifBar

Video 42: Human9

Video 43: Walking2

Video 44: Skiing



Video 45: Human4

Video 46: Skating2-1

Video 47: Skating2-2

Video 48: Walking

Video 49: Crowds

Video 50: Diving

The references used in the above comparisons:

[27] Henriques J F, Caseiro R, Martins p, Batista J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015, 37(3): 583–596.

[39] He S, Yang Q X, Lau R, Wang J, Yang M H. Visual tracking via locality sensitive histograms, Proc of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, 2013: 2427–2434.

[13] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark, Proc of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, 2013: 2411–2418.

[29] Ning J, Yang J, Jiang S, Zhang L, Yang M H. Object tracking via dual linear structured SVM and explicit feature map, Proc of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 4266–4274.

[5] Hare S, Saffari A, Torr P H S. Efficient online structured output learning for key point-based object tracking, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Providence, 2012: 1894–1901.

[32] Li Y, Zhu J, Hoi S C H. Reliable patch trackers: robust visual tracking by exploiting reliable patches, Proc of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015:353–361.

[9] Hare S, Saffari A, Torr P H S. Struck: Structured output tracking with kernels, IEEE Transactions on Pattern Recognition and Machine Intelligence, 2016, 38(10): 2096–2109.

[3] Jia X, Lu H, Yang M H. Visual tracking via adaptive structural local sparse appearance model, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, 2012:1822–1829.

[12] Zhong W, Lu H, Yang M H. Robust object tracking via sparsity-based collaborative model, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Providence (CVPR), 2012: 1838–1845.

Appendix 2: The Evaluations on 50 videos of TB50 from Tracking Benchmark

8 trackers are used to do comparisons, including the recent famous trackers DLSSVM (CVPR'16) [29], KCF(PAMI' 15) [27], RPT (CVPR' 15) [32], LSHT(CVPR '13) [39], LSST(CVPR' 13) [5] and the top three ranked trackers from Tracking Benchmark [13] namely the STRUCK (PAMI'16) [9], ALSA (CVPR'12) [3], SCM (CVPR'12) [12]. We select 6 trackers (RPT, DLSSVM, KCF, LSHT, LSST) to demonstrate the results, and they perform better in the compared 8 trackers and proposed in recent years. The references of these trackers are described at the end of this file.

The precision plots of 11 tracking challenges on TB50:

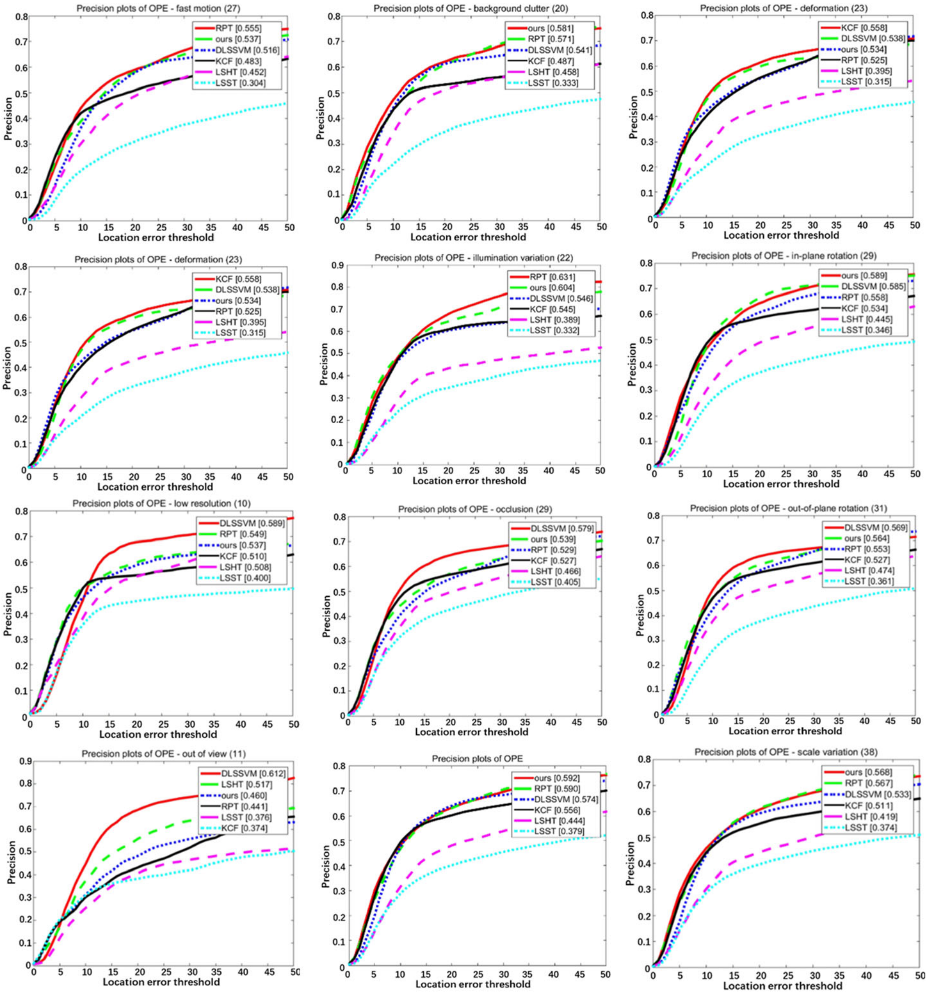


Fig. 11 The precision plots of 11 tracking challenges on TB50

The success plots of 11 tracking challenges on TB50:

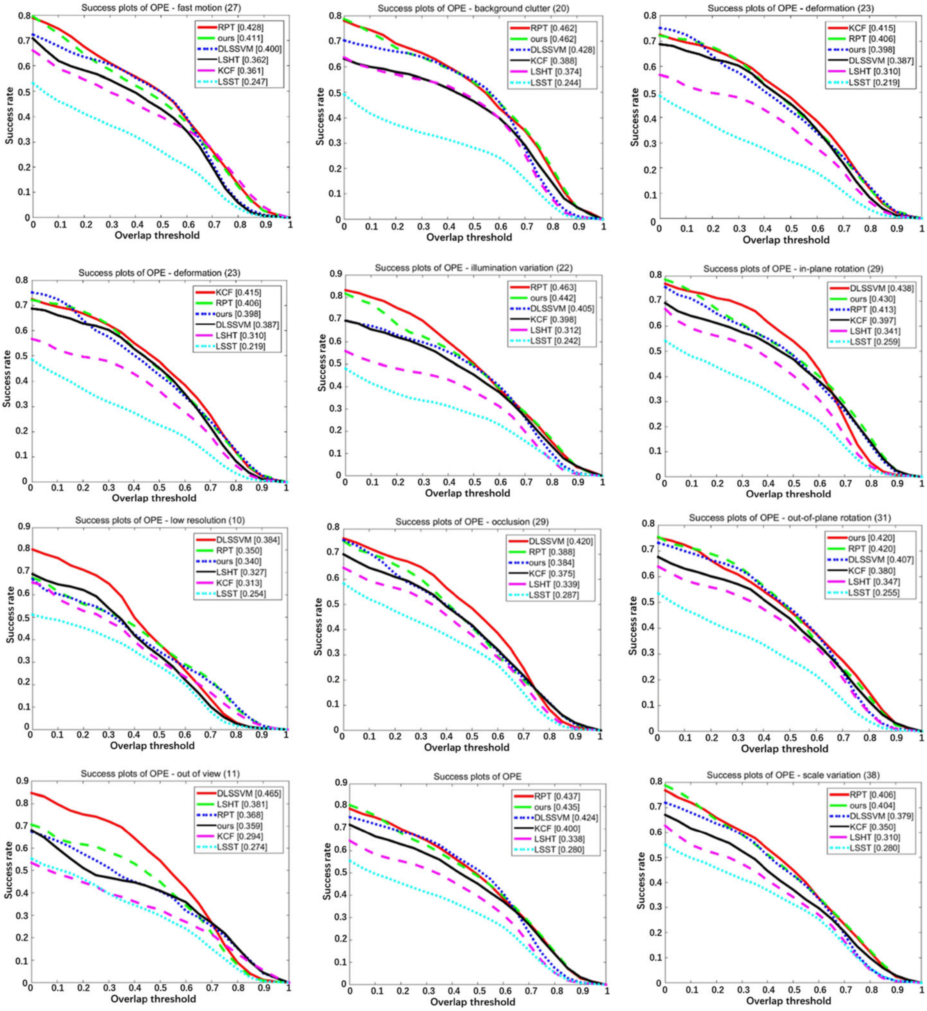


Fig. 12 The precision plots of 11 tracking challenges on TB50

The references used in the above comparisons:

[27] Henriques J F, Caseiro R, Martins p, Batista J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015, 37(3): 583–596.

[39] He S, Yang Q X, Lau R, Wang J, Yang M H. Visual tracking via locality sensitive histograms, Proc of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, 2013: 2427–2434.

[13] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark, Proc of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland, 2013: 2411–2418.

[29] Ning J, Yang J, Jiang S, Zhang L, Yang M H. Object tracking via dual linear structured SVM and explicit feature map, Proc of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 4266–4274.

[5] Hare S, Saffari A, Torr P H S. Efficient online structured output learning for key point-based object tracking, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Providence, 2012: 1894–1901.

[32] Li Y, Zhu J, Hoi S C H. Reliable patch trackers: robust visual tracking by exploiting reliable patches, Proc of the 29th IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015:353–361.

[9] Hare S, Saffari A, Torr P H S. Struck: Structured output tracking with kernels, IEEE Transactions on Pattern Recognition and Machine Intelligence, 2016, 38(10): 2096–2109.

[3] Jia X, Lu H, Yang M H. Visual tracking via adaptive structural local sparse appearance model, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, 2012:1822–1829.

[12] Zhong W, Lu H, Yang M H. Robust object tracking via sparsity-based collaborative model, Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Providence (CVPR), 2012: 1838–1845.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Bibi A, Mueller M, Ghanem B. Target response adaptation for correlation filter tracking, proc of the 14th European conference on computer vision. Amsterdam, 2016: 419–433
2. Bolme DS, Beveridge JR, Draper B, Lui YM et al. (2010) Visual object tracking using adaptive correlation filters. Proc 23th IEEE Conf Comput Vision Pattern Recogn (CVPR) San Francisco: 2544–2550
3. Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters, proc of the 23th IEEE conference on computer vision and pattern recognition (CVPR). San Fracisco: 2544–2550
4. Chen D, Yuan Z, Wu Y, Zhang G, Zheng N (2013) Constructing adaptive complex cells for robust visual tracking. Proc 19th Int Conf Comput Vision. Sydney:1113–1120
5. Comaniciu D, Ramesh V, Meer P Kernel-based object tracking, IEEE Trans Pattern Anal Mach Intell 2003, 25 (5): 564–575
6. Cui Z, Xiao S, Feng J, Yan S (2016) Recurrently target-attending tracking. Proc 29th IEEE Conf Comput Vision Pattern Recogn (CVPR). Las Vegas: 1449–1458
7. Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. Proc 21th Int Conf Comput Vision (ICCV). Santiago: 4310–4318
8. Fan H, Ling H (2017) SANet: structure-aware network for visual tracking. Proc 30th IEEE Conf Comput Vision Pattern Recogn (CVPR), Hawaii: 2217–2224
9. Godec M, Roth PM, Bischof H (2013) Hough-based traking of non-rigid objects. Comput Vis Image Underst 117(10):1245–1256
10. Hamed KG, Ashton F, Simon L (2017) Learning background-aware correlation filters for visual tracking. Proc 22th IEEE Conf Int Conf Comput Vision (ICCV), Venice: 1144–1152
11. Hare S, Saffari A, Torr PHS (2012) Efficient online structured output learning for key point-based object tracking. Proc 25th IEEE Conf Comput Vision Pattern Recogn. Providence: 1894–1901
12. Hare S, Saffari A, Torr PHS (2016) Struck: structured output tracking with kernels. IEEE Trans Pattern Recogn Mach Intell 38(10):2096–2109
13. He S, Yang QX, Lau R, Wang J, Yang MH (2013) Visual tracking via locality sensitive histograms. Proc 26th IEEE Conf Comput Vision Pattern Recogn (CVPR). Portland: 2427–2434
14. Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. Proc 12th Eur Conf Comput Vision. Florence: 702–715

15. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
16. Hu Z, Xie R, Wang M, Sun Z (2017) Midlevel cues mean shift visual tracking algorithm based on target-background saliency confidence map. Multimed Tools Appl 76:21265–21280
17. Jack V, Luca BF, Joao FH, Andrea V, Philip HST (2017) End-to-end representation learning for correlation filters based tracking. Proc 30th IEEE Conf Comput Vision Pattern Recogn (CVPR), Hawaii: 5000–5008
18. Jia X, Lu H, Yang MH (2012) Visual tracking via adaptive structural local sparse appearance model. Proc 25th IEEE Conf Comput Vision Pattern Recogn. Providence:1822–18292
19. Jia X, Lu H, Yang MH (2012) Visual tracking via adaptive structural local sparse appearance model, proc of the 25th IEEE conference on computer vision and pattern recognition (CVPR). Providence:1822–1829
20. Jongwon C, Hyung JC, Sangdoo Y, Tobias F (2017) Attentional correlation filter network for adaptive visual tracking, proc of the 30th IEEE conference on computer vision and pattern recognition (CVPR), Hawaii: 4828–4837
21. 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
22. Li Y, Zhu J, Hoi SCH (2015) Reliable patch trackers: robust visual tracking by exploiting reliable patches. Proc 29th IEEE Conf Comput Vision Pattern Recogn. Boston:353–361
23. Liao L (2017) X, Zhang C, toward situation awareness: a survey on adaptive learning for model-free tracking. Multimed Tools Appl 76:21073–21115
24. Liu Y, Cui J, Zhao H, Zha H (2012) Fusion of low-and high-dimensional approaches by trackers sampling for generic human motion tracking. Proc 21st Int Conf Pattern Recogn (ICPR), Japan, Tsukuba Science, , 898–901
25. Liu Y, Nie L, Han L, Zhang L, Rosenblum DS (2015) Action2activity: Recognizing complex activities from sensor data. Proc 24th Int Conf Artif Intell (IJCAI), Buenos Aires, Argentina: 1617–1623
26. Liu Y, Nie L, Liu L, Rosenblum DS (2016) From action to activity: Sensor-based activity recognition. Neurocomputing 181(12):108–115
27. Liu S, Zhang T, Cao X, Xu C (2016) Structural correlation filter for robust visual tracking. Proc 29th IEEE Conf Comput Vision Pattern Recogn (CVPR). Las Vegas: 4312–4320
28. Liu Y, Zhang L, Nie L, Yan Y, Rosenblum DS (2016) Fortune teller: predicting your career path. Proc thirtieth AAAI Conf Artif Intell (AAAI), Phoenix, Arizona: 201–207
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
30. Ma C, Huang JB, Yang X, Yang MH (2015) Hierarchical convolutional features for visual tracking. Proc 21th Int Conf Comput Vision (ICCV). Santiago: 3074–3082
31. Martin D, Goutam B, Fahad K, Michael F (2017) ECO: efficient convolution operators for tracking. Proc 30th IEEE Conf Comput Vision Pattern Recogn (CVPR), Hawaii: 6931–6939
32. Mohanapriya D, Mahesh K (2017) A novel foreground region analysis using NCP-DBP texture pattern for robust visual tracking. Multimed Tools Appl 76:25731–25748
33. Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. Proc 29th IEEE Conf Comput Vision Pattern Recogn (CVPR) Las Vegas : 4293–4302
34. Ning J, Yang J, Jiang S, Zhang L, Yang MH (2016) Object tracking via dual linear structured SVM and explicit feature map. Proc 29th IEEE Conf Comput Vision Pattern Recogn. Las Vegas 4266–4274
35. Pan Z, Liu S, Fu W (2017) A review of visual moving target tracking. Multimed Tools Appl 76:16989–17018
36. Quan W, Liu Z, Chen JX, Liang D (2017) Adaptive relay detection using primary and auxiliary detectors for tracking. Multimed Tools Appl 76:24299–24313
37. Smeulders AWM, Chu DM, Calderara S, Dehghan A, Shah M (2014) Visual tracking: an experiment survey. IEEE Trans Pattern Anal Mach Intell 36(7):1442–1468
38. Wang L, Ouyang W, Wang X, Lu H (2016) STCT: sequentially training convolutional networks for visual tracking. Proc 29th IEEE Conf Comput Vision Pattern Recogn (CVPR). Las Vegas: 1373–1381
39. Wang Z, Wang H, Tan J, Chen P, Xie C (2017) Robust object tracking via multi-scale patch based sparse coding histogram. Multimed Tools Appl 76:12181–12203
40. Wang M, Liu Y, Huang Z (2017) Large margin object tracking with circulant feature maps. Proc 30th IEEE Conf Comput Vision Pattern Recogn (CVPR), Hawaii: 4800–4808
41. Wang F, Li X, Lu M (2017) Adaptive Hamiltonian MCMC sampling for robust visual tracking. Multimed Tools Appl 76:13087–13106
42. Wu Y, Lim J, Yang MH (2013) Online object tracking: a benchmark. Proc 26th IEEE Conf Comput Vision Pattern Recogn (CVPR). Portland: 2411–2418
43. Xu Y, Cui J, Zhao H, Zha H (2013) Tracking generic human motion via fusion of low-and high-dimensional approaches. IEEE Trans Syst Man Cybernet: Syst 43(4):996–1002
44. Yang F, Lu H, Yang MH (2014) Robust superpixel tracking. IEEE Trans Image Process 23(4):1639–1651

45. Yun S, Choi J, Yoo Y, Yun K, Choi Y (2017) Action-decision networks for visual tracking with deep reinforcement learning. Proc 30th IEEE Conf Comput Vision Pattern Recogn (CVPR), Hawaii: 1349–1358
46. Zhang L, Maaten L (2014) Preserving structure in model-free tracking. IEEE Trans Pattern Recogn Mach Intell 36(4):756–769
47. Zhong W, Lu H, Yang MH (2012) Robust object tracking via sparsity-based collaborative model, proc of the 25th IEEE conference on computer vision and pattern recognition. Providence (CVPR): 1838–1845

Yun Liang is an Associated 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.

Ke Li received the M.Sc. and Ph.D. degrees from the Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China, in 2005 and 2008, respectively. He is currently an Associated Professor with the Zhengzhou Institute of Surveying and Mapping. His research interests include geographic information system, deep learning, and remote sensing image analysis.

Jian Zhang received the Ph.D. degree from Zhejiang University, Zhejiang, China. He is currently an Associated Professor with the School of Science and Technology, Zhejiang International Studies University, Hangzhou, China. From 2009 to 2011, he was with Department of Mathematics of Zhejiang university as a Post-doctoral Research Fellow. In 2016, he had been doing research on machine learning at Simon Fraser University (SFU) as a Visiting Scholar. His research interests include but not limited to machine learning, computer animation and image processing. He serves as a reviewer of several prestigious journals in his research domain.

Mei-hua Wang is an Associated 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.

Affiliations

Yun Liang[1] · **Ke Li**[2] · **Jian Zhang**[3] · **Meihua Wang**[1] · **Chen Lin**[4]

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

[2] Zhengzhou School for Surveying and Mapping, Zhengzhou 450052, China

[3] School of Science and Technology, Zhejiang International Studies University, Hangzhou 310012, China

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