Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
November-December 2019: 1–13
© The Author(s) 2019
DOI: 10.1177/1729881419890155
journals.sagepub.com/home/arx
SAGE

Incremental learning-based visual tracking with weighted discriminative dictionaries

Penggen Zheng[1], Huimin Zhao[1], Jin Zhan[1], Yijun Yan[2], Jinchang Ren[2], Jujian Lv[1] and Zhihui Huang[1]

Abstract

Existing sparse representation-based visual tracking methods detect the target positions by minimizing the reconstruction error. However, due to complex background, illumination change, and occlusion problems, these methods are difficult to locate the target properly. In this article, we propose a novel visual tracking method based on weighted discriminative dictionaries and a pyramidal feature selection strategy. First, we utilize color features and texture features of the training samples to obtain multiple discriminative dictionaries. Then, we use the position information of those samples to assign weights to the base vectors in dictionaries. For robust visual tracking, we propose a pyramidal sparse feature selection strategy where the weights of base vectors and reconstruction errors in different feature are integrated together to get the best target regions. At the same time, we measure feature reliability to dynamically adjust the weights of different features. In addition, we introduce a scenario-aware mechanism and an incremental dictionary update method based on noise energy analysis. Comparison experiments show that the proposed algorithm outperforms several state-of-the-art methods, and useful quantitative and qualitative analyses are also carried out.

Keywords

Visual tracking, similarity weights, sparse representation, incremental update, weighted dictionary

Date received: 1 February 2019; accepted: 25 October 2019

Topic: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Kaifu Yang

Introduction

As a subtask of computer vision, visual target tracking has always drawn many attentions for decades, and many advanced methods have been explored. However, complex situations such as occlusions, target deformation, rotation, scale changes, and cluttered background, and so on, make visual target tracking still a challenging task and the existing methods cannot always track the targets precisely. The current trackers can be typically divided into two types, that is, generative methods[1–5] and discriminative methods.[6–17] They usually sample a set around the target object to describe the appearance characteristics, and search for

[1] School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China
[2] Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK

Corresponding author:
Jin Zhan, School of Computer Science, Guangdong Polytechnic Normal University, Tianhe District Zhongshan Road West, No. 293 Guangzhou, Guangdong 510665, China.
Email: gszhanjin@gpnu.edu.cn

candidate targets by maximizing the similarity or find the decision boundaries of the target and the background.

To get the satisfied tracking performance, two key issues need to be addressed. First, since the appearance of the target changes frame by frame throughout the video sequence, the most discriminating samples in the current frame may not last for a long time and tend to result in a model overfitting. So, the improvement of long-term tracking performance is an important issue. Second, Unpredictable target deformation and background clutter in the sampling region cause a negative impact on the selection of candidate samples. Thus, the elimination of these obstacles to advance tracking performance in the case of small target samples is also an important issue.

To address both issues, sparse representation-based tracking solutions have been proposed, such as L1 tracker.[5] Because of insensitivity to the target noise, this kind of methods has a strong tracking robustness when target deformation occurs. However, single-feature and the initial discriminative dictionary do not satisfy complex tracking scenarios. Moreover, the object localization under the frequent online updating often brings drift-away problems as some negative samples are mis-tracked. These problems remain difficult in the literature of sparse representation-based trackers. Hence, a natural question is how we can augment positive samples in the feature space to capture target appearance variations in the temporal domain.

In this work, we take advantage of the recent progress in discriminative dictionary learning method label consistent K-SVD (LC-KSVD)[18,19] to facilitate the dictionary learning and to propose a novel tracking method with weighted dictionaries incremental learning and pyramidal feature selection strategy. In summary, this work has the following main steps. Firstly, we model the discriminative dictionaries from positive and negative samples based on two feature descriptors, where different features correspond to different dictionaries. Secondly, according to the center distance from the training samples to the target, we assign Gaussian weights for each basis vector in different feature dictionaries, which are used to measure the similarity of spatial structure to improve the accuracy of sparse feature selection. Finally, we select the best sample region by similarity measurement and fusion of the multiple features reconstruction error of candidate samples.

The article is organized as follows. We introduce the research background in the "Introduction" section and review the related work in the "Related work" section. Afterwards, the "Proposed method" section describes the proposed method in detail, including dictionary representation and construction, incremental dictionary updating, and adaptive feature fusion strategy. The experiments are given in "Experimental results and comparison." We conclude the article and discuss future work in the "Conclusion and future work" section.

Related work

In this section, we briefly review the relevant literature of object tracking algorithms in recent year, including deep learning-based tracking methods[11–17,20,21–23,24] and sparse representation-based tracking method.[1,3,5,6,25–28,30,31]

The main advantage of deep learning-based tracking methods lies in their powerful characterization of depth features. It brings a new research direction for solving various challenges in visual tracking. Wang and Yeung[20] proposed deep learning tracker and performed unsupervised off-line depth pretraining on large-scale natural image data sets. The idea of transfer learning reduces the requirement of training samples and improves the performance of the tracking algorithm. Then, they propose structured output-deep learning tracker[11] and use convolutional neural network (CNN) model to solve the sensitivity of model updating. Qi et al.[12] proposed a novel CNN-based tracking method, which considers the features from all CNN layers and hedge these features into a single stronger one. Furthermore, they propose a hedging deep feature-based tracking framework[13] which use correlation filters to feature maps of each CNN layer to construct a weak tracker and design a Siamese network to define the loss of each weak tracker. The tracker achieves favorable performance on challenging image sequences.

To solve the imbalance between positive and negative samples in video tracking, Zhang et al.[14] proposed an attribute-based CNN with multiple branches, where each branch is responsible for classifying the target under a specific attribute. The tracker reduces the appearance diversity of the target under each attribute and thus requires fewer data to train the model. Qi et al.[15] proposed to integrate the point-to-set/image-to-imageSet distance metric learning (DML) into visual tracking. The point-to-set DML is conducted on CNN features of the training data, and the tracking result is located by the minimal distance to the target template. Because the methods based on matching tracking cannot deal with the problem of target rotation in the plane very well, Zhong et al.[16] proposed a hierarchical tracker that learns to move and track by a coarse-to-fine verification. The coarse level utilizes a recurrent CNN-based deep Q-network to learn data-driven searching policies. The idea of learning target position from coarse to fine is helpful to deal with target scale change and improve the accuracy of tracking target border. The authors also apply this idea to multi-person tracking and propose a deep alignment network-based multi-person tracking method[17] with occlusion and motion reasoning which achieves good performance. Wang et al.[24] proposed a deep learning-based hybrid

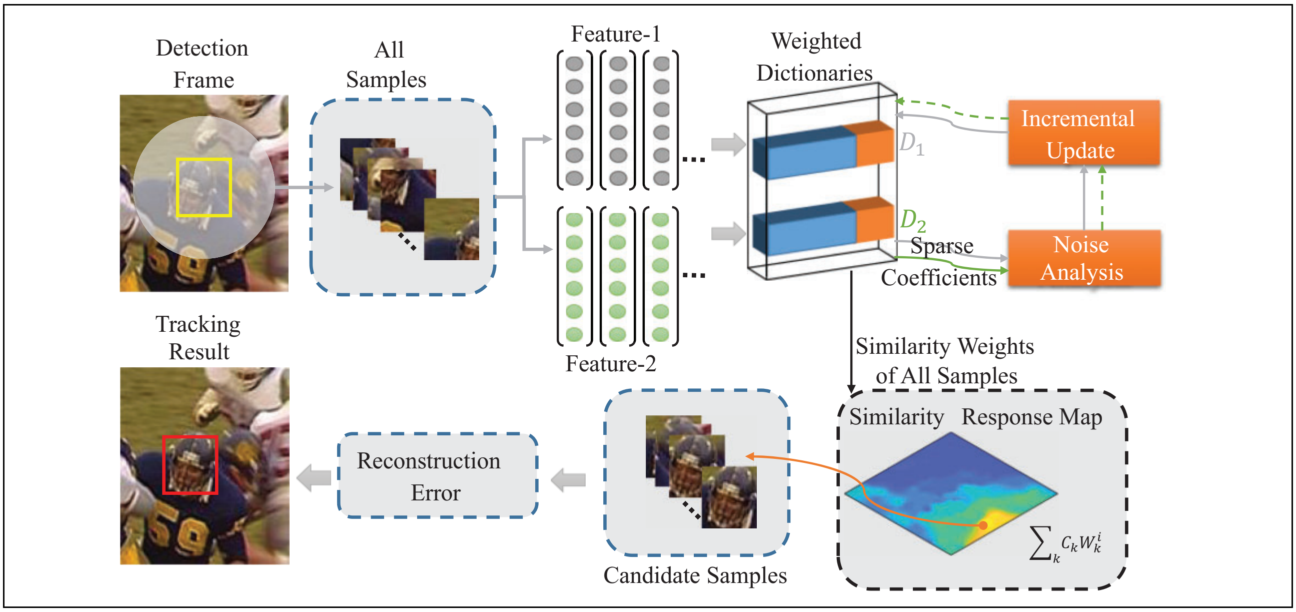

Figure 1. The main tracking process of the proposed approach.

spatiotemporal saliency feature extraction framework for saliency detection from video footages.

Sparse representation-based tracking methods show strong robustness in some tracking scenarios. Therefore, many visual tracking methods[5,25,27,31,32] based on sparse representations have been proposed. Local sparse representations are widely used in visual tracking.[28,29,33] Zhang et al.[30] summarized and evaluated some classical tracking methods based on sparse representation. The process of sparse representation-based trackers can be roughly divided into two stages. The first stage acquires a sparse sample set around the target, and the second stage uses a classifier to classify each sample as a target or background. However, the positive samples obtained from the first frame of video are far from meeting the requirement of label data volume in classifier training, and the positive and negative samples are imbalanced greatly, which makes it impossible to capture the rich appearance changes of the target. These limitations are also reflected in some deep learning-based trackers[21–23] that use this two-stage framework.

In the target tracking process, a good model update strategy can improve the tracking effect and tracking ability. Lu et al.[26] used incremental subspace learning methods to reconstruct a new template and then utilized it to replace the old one. However, the updated base vector will gradually degrade in the scene where noise or occlusion exists. In addition, Mei and Ling[5] replaced the least important template with the current template based on the frequency of use of the dictionary template. Han et al.[27] updated the dictionary template in a random replacement manner.

The combination of multiple features enhances the characterization capabilities of the model and is applied to many different classification tasks. From the perspective of visual attention saliency, Yan et al.[34,35] used Gestalt rule to guide the saliency detection by characterizing human visual system (HVS) features and forming targets and proposed a method to cognitively detect and track salient objects from videos by combining red-green-blue (RGB) image and thermal image. The proposed fusion-based approach can successfully detect and track multiple human objects in most scenes regardless of any light change or occlusion problem. Lan et al.[25] used an unreliable feature detection method to detect unreliable features. However, the representation of reliable features is still suppressed by the joint sparse framework, and different features are limited to similar sparse patterns. Mai and Ling[5] fused multiple features for appearance modeling and detect the outlier particles. The same sparse pattern is used for all features of the non-outlier particles.

In this article, we propose a novel multifeatures dictionary-based sparse tracking method, where a specific feature dictionary is built upon hybrid features with the ability of independently maintaining. Then an incremental dictionary update strategy is proposed to reduce the redundancy of sparse dictionaries while increasing the diversity of positive samples. The output of these dictionaries responses in a different sparse pattern for the final comprehensive decision during the tracking process.

Proposed method

In this section, the proposed method including three modules is introduced. The main framework of our method is shown in Figure 1. We maintain two sets of samples (positives and negatives) to construct weighted feature

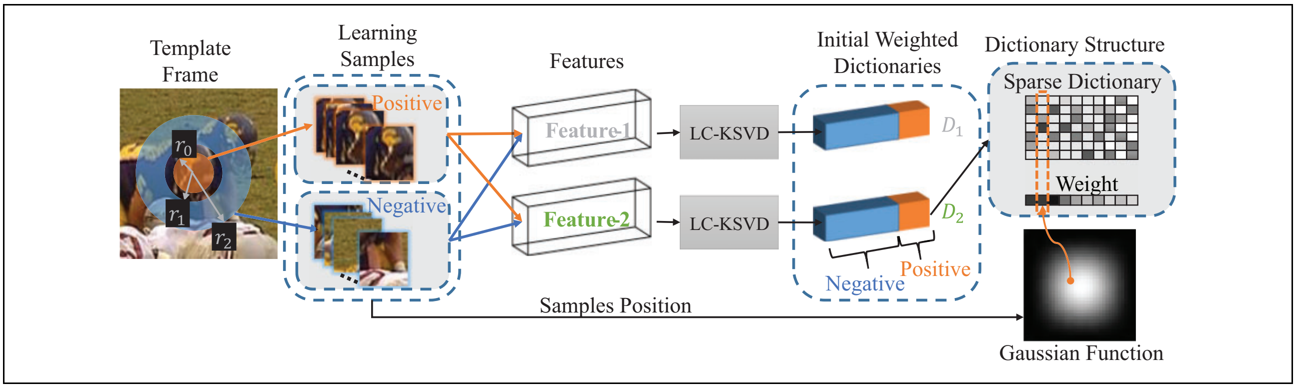

Figure 2. Initial dictionary learning and the structure of multiple dictionaries.

dictionaries. In the tracking process, the samples are sparsely decomposed by the weighted dictionaries, and the weights of the samples can be obtained and used to select candidate samples. By comparing the reconstruction errors of these candidate samples, we can select the most similar sample as the tracking result.

Dictionary representation and construction

In sparse representation theory, dictionary is composed of super-complete base vectors to obtain a more concise representation of the appearance of the target. For this purpose, three types of sets, that is, the positives T, the backgrounds B, and the noise L are integrated together. The initial dictionary D of the samples at the first frame can be represented as $D = [D^T, D^B, D^L]$, where D^T, D^B, and D^L are the sets of T, B, and L, respectively. In the tracking process, a candidate sample y can be represented by the sparse representation (equation (1))

$$y \approx D\gamma = \left[D^T, D^B, D^L\right] \begin{bmatrix} z \\ v \\ e \end{bmatrix} \tag{1}$$

where D is the discriminative dictionary, z is the target coefficient, v is the background coefficient, e is the noise coefficient, and γ is the sparse coding. In this article, the LC-KSVD[18] method is used to unify dictionary learning and classification labeling.

Figure 2 shows the construction process of the initial dictionary. The center of the initial target is set as the center of the circle, pixels in the range of radius r_0 are sampled as positive samples, and pixels in the range of radius between r_1 and r_2 are dense sampled to obtain negative samples which contain the background context around the target.

For the positive and negative samples sampled in the first frame, we extract two kinds of features to form two initial dictionaries respectively. After that, we utilize the correspondence between the sample template and the dictionary base vector and assign the Gaussian weight to each base vector by calculating the center distance $d(i)$ between sample templates and the target center. The weight of the ith base vector is defined as follows

$$W(i) = \exp(-d^2(i)/2\sigma^2) \tag{2}$$

where α is the standard deviation of normal distribution. This weight reflects the similarity between the target and the samples. Finally, we get the weighted discriminative dictionaries, and each discriminative dictionary corresponds to a weight table.

Incremental dictionary updating

In many existing tracking methods, the appearance model of target is often updated to reduce the negative impact of target and background changes in the frames. In the sparse decomposition, the coefficient γ of sample contains the most representative information, where the noise factor indicates the situation of target occlusion and tracking drift to some extent. To this end, an incremental dictionary updating strategy is proposed to measure the change of target or scene by analyzing the noise energy u (the sum of the noise coefficients e). The larger the noise energy is, the more significant the deformation of the target or the greater the change of the scene causes.

In the frame t, the average noise energy expression for all samples can be represented as $\overline{u_{k,t}} = \frac{\sum_i u_k^i}{n}$, where u_k^i is the noise energy of the ith sample, n is the number of all samples, and k is the feature tag ($k = 1$ denotes color feature and $k = 2$ denotes noncolor feature). We define a dynamic threshold to analyze changes of the target and scenes, which are defined as follows

$$P\{U_k > x_k^\alpha\} = \alpha \tag{3}$$

where x_k^α is the upper quantile of set U_k (all $\overline{u_{k,t}}$ from the first frame to current frame) and reflects the overall level of noise energy during the tracking process. If the tracked average noise energy $\overline{u_k^T}$ exceeds the threshold, it indicates that the background changes too much in the

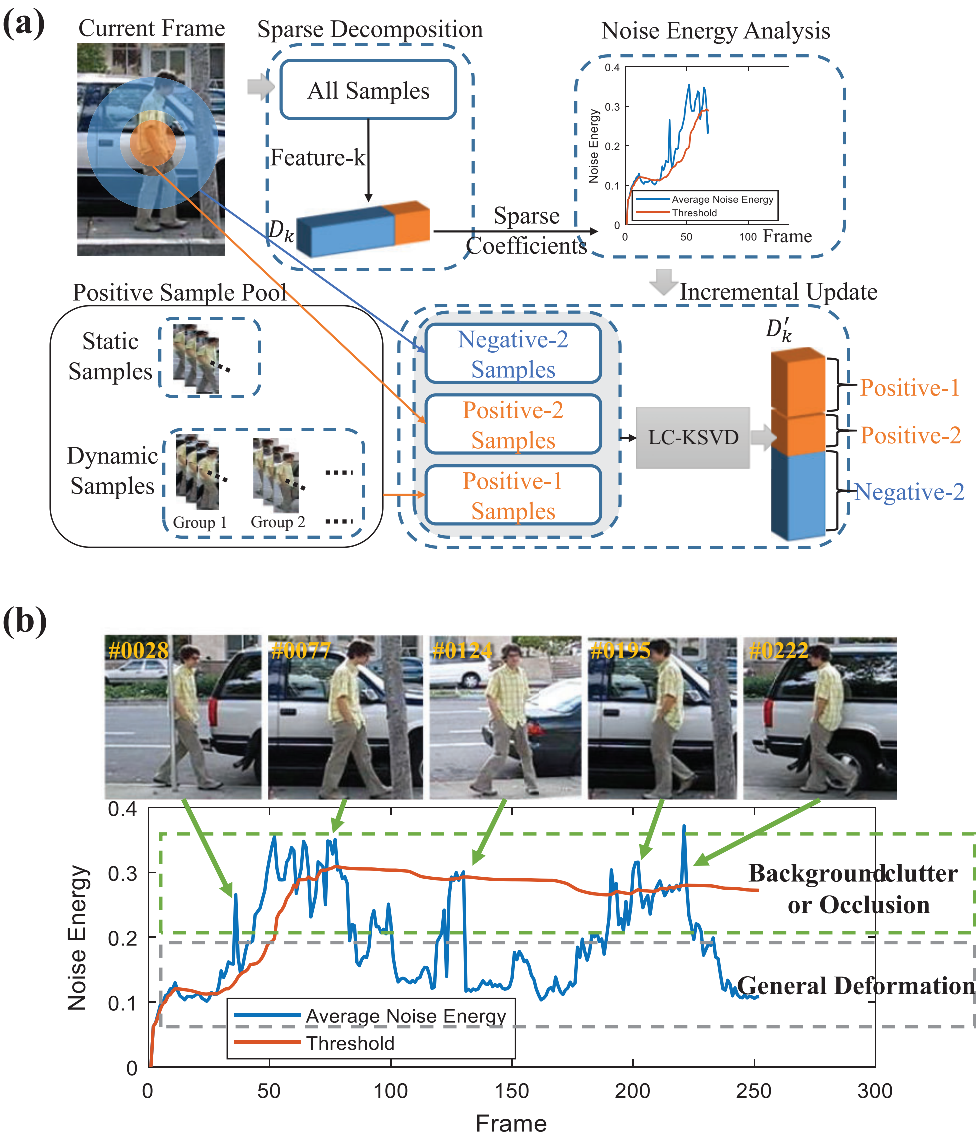

Figure 3. Discriminative dictionary incremental update (a) and noise analysis (b).

current frame. Meanwhile, we also set the minimum update interval m to make the tracking process more efficient and the interval between two updates must be more than m frames.

In scene detection, we use the dynamic threshold of noise energy to judge the intensity of the scene change. Based on the target noise energy and the average noise energy, we can determine whether to perform a dictionary update. If the update condition is met, we use the samples of the first frame and the samples of the detected frame to obtain a new weighted dictionary D_k'. The new dictionary will be used for the next frame tracking task.

The incremental dictionary update trigger mechanism is shown in Figure 3(a). We divide the positive samples into two categories: static samples and dynamic samples. The samples obtained in the first frame are static samples, and the samples obtained in the trigger update mechanism are dynamic samples. When the number of positive samples is larger than that of current negative samples, we use a new positive sample set to randomly replace one group of the dynamic samples to reduce the impact of sample imbalance and maintain the efficiency of dictionary learning.

Figure 3(b) shows the changes of threshold curve and noise energy curve in sequence *David 3*. Five frames with large changes of target pose and background are selected as examples for illustration. It can be seen that the selected examples occur when the noise energy value is higher than the threshold value. Hence, our updating strategy can detect and reduce the impact of the background change in real time through the analysis of noise energy for better tracking performance.

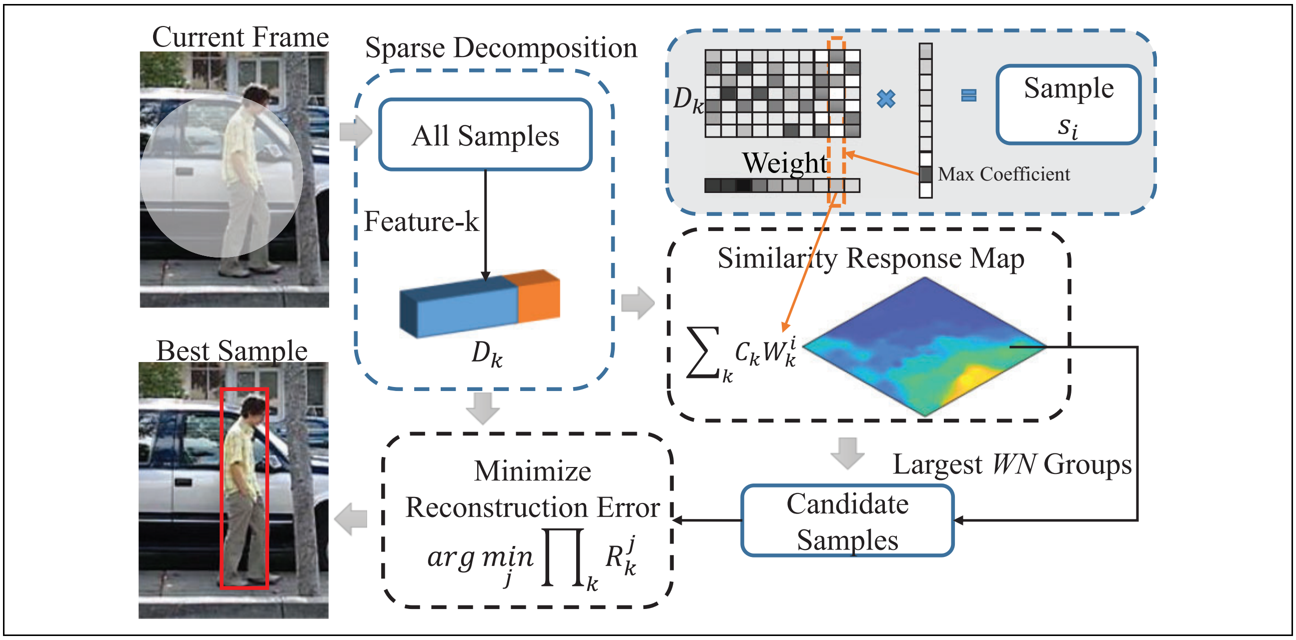

Figure 4. Feature selection process.

Adaptive feature fusion strategy

In this section, we introduce the pyramid feature selection strategy to locate the target tracking position, as shown in Figure 4. We use a pyramidal selection strategy in the feature selection. First, we select WN groups of samples with the largest similarity weights as candidate samples CS_j (j is the tag of the candidate samples). The sample similarity weight can be obtained in the sparse decomposition process. Then, we compare the comprehensive reconstruction error of the candidate samples to select the best sample as the tracking result.

In the current frame, all samples S_i ($i = 1, 2, \ldots, n$) are sparsely resolved by different feature dictionaries D_k ($k = 1, 2$) to obtain sparse coefficients γ_k^i, where k is a feature tag. The similarity weights W_k^i and reconstruction errors R_k^j are normalized into [0,1] to eliminate the inconsistency of different feature weights. Each sample S_i has k feature sparse coefficients. The weight values corresponding to the maximum values of the k feature sparse coefficients are used as the similarity weights W_k^i ($k = 1, 2$). Therefore, we fuse these two weights into a composite weight w^i, which is defined as follows

$$tW^i = \sum\nolimits_k C_k W_k^i \tag{4}$$

$$C_k = 1 - \frac{\overline{u_k^T}/x_k^\alpha}{\sum_{k=1}^{2} \overline{u_k^T}/x_k^\alpha} \tag{5}$$

In equation (4), we set the dynamic feature weight parameters C_k based on the feature reliability. Then we select a few candidate samples CS_j which have the largest synthetic weights among all samples and the maximum value of synthetic weights is denoted as WN. When the noise energy is relatively large, the feature weight C_k is relatively small. The definition of C_k is shown in equation (5). $\overline{u_k^T}$ is the k-feature average noise energy of the current frame, and x_k^α is the k-feature noise energy threshold defined in equation (3).

Then we use the synthetic reconstruction error to select the best sample from candidate samples. The expression of the synthetic reconstruction error is as follows

$$R^j = \prod\nolimits_k R_k^j \tag{6}$$

where R_k^j represents the reconstruction error of the sample s_j in *k-feature*, j is the label of the candidate samples. Finally, we select the one with the smallest synthetic reconstruction error in the candidate samples as our tracking result.

Experimental results and comparison

In this section, the public sequences of VOT2017[36] and OTB100[37] are used for the parameter setting and tracking performance evaluation of our method, respectively. Firstly, we experiment with eight RGB sequences of VOT2017,[36] analyze the parameter settings in the feature selection, and discuss the optimal combination of features. Then all 74 RGB sequences on the OTB100[37] are used for tracking performance evaluation. The experiment tracking results of other benchmarking methods are primarily derived from publicly available results data on the author's homepage and OTB100[37] homepage. The computer environment used by our method is Intel (R) Core (TM) i3-3.7 GHz, RAM-12 GB, and MATLAB R2017a.

Table 1. The average CLEs for different dual feature combinations.[a]

Type	Feature (s)	*ball1*	*blanket*	*butterfly*	*crossing*	*godfather*	*pedestrian1*	*sheep*	*wiper*	Average
Color feature	HSV	5.69	10.42	21.22	47.57	15.06	41.18	43.66	24.46	26.16
	RGB	43.8	17.85	27.21	40.75	16.89	21.71	35.46	216.36	52.5
	LAB	4.91	12.76	19.74	45.48	10.35	13.41	43.11	163.71	39.18
Noncolor feature	Haar-like	12.4	49.04	76.42	37.05	9.27	96.55	**6.48**	25.68	39.11
	HOG	62.21	40.11	41.9	19.67	19.56	66.26	93.36	70.48	51.69
Fusion of color feature and noncolor feature	HOG + HSV	7.15	**9.54**	22.15	**17.91**	10.13	12.08	76.79	**15.61**	21.42
	HOG + RGB	29.87	9.85	30.13	19.53	7.95	13.54	22.24	43.33	22.06
	HOG + LAB	5.28	9.75	19.77	30.41	8.65	**10.39**	41.28	81.83	25.92
	Haar-like + HSV	3.69	11.05	27.97	42.84	7.14	23.26	11.58	19.05	**18.32**
	Haar-like + RGB	4.07	16.31	44.41	28.99	**6.89**	22.14	12.38	36.81	21.5
	Haar-like + LAB	3.23	14.92	27.4	35.49	7.52	15.57	39.81	29.08	21.63

CLE: center location error.

[a] Bold data represent the best results of single video tasks.

Table 2. The average CLEs with different WN values.[a]

Sequences	WN = 1	WN = 2	WN = 3	WN = 4	WN = 5	WN = 6	WN = 7
ball1	5.066321	5.051257	4.621554	3.477649	4.148773	4.210412	3.893678
blanket	18.31654	10.40168	10.52438	11.24985	11.77508	15.84256	15.01424
butterfly	27.08312	28.87616	28.02551	30.88001	30.57644	30.84231	30.94732
crossing	45.38556	46.61894	47.04421	37.7282	42.01624	37.27262	35.52469
godfather	9.649543	7.13208	8.276668	7.635623	12.19341	7.387032	7.490803
pedestrian1	19.45411	20.88936	21.5531	19.13769	34.73961	15.42067	29.3694
sheep	35.82426	30.52743	22.04572	25.95596	27.32844	45.41071	22.90445
wiper	20.25632	24.19629	22.58765	20.48694	25.05795	26.48748	19.63991
Average	22.62947	21.71165	20.58485	19.56899	23.47949	22.85922	20.59806

CLE: center location error.

[a] *Source*: The parameter setting of the variable WN is from 1 to 7.

Implementation details and analysis

The method of this article adopts uniform parameter settings. The number of all samples obtained by Gaussian sampling during the tracking process is 500 and the sampling radius is 25. The sampling parameter of the training sample is set to: $r_0 = 4$, $r_1 = 7$, $r_2 = 15$. The Haar-like[38] feature dimension is set to 150, and the histogram bin of a single color channel is set to 36. Correspondingly, the color feature dimension of an RGB frame is set to 108. The update time interval must be greater than $m = 6$ frames, and the noise energy threshold parameter is $\alpha = 0.2$.

Feature selection. Two types of features (noncolor features and color features) are used in our proposed model. In this section, the performance of different feature fusion strategy on eight RGB sequences (*ball1*, *blanket*, *butterfly*, *crossing*, *godfather*, *pedestrian1*, *sheep*, and *wiper*) in VOT2017[32] is investigated and useful analysis is also carried out.

Table 1 shows the performance of different feature fusion strategies in terms of average center location errors (CLEs). The CLE is the Euclidean distance between the tracking result and the standard target position. In general, dual feature fusion always outperforms single feature. Feature CIE L*a*b* (LAB) performs poorly in combination with other non-color features. It is worth noting that histogram of orientation gradient (HOG) + hue-saturation-value (HSV) has the best performance in the sequences of *blanket*, *crossing*, and *wiper*, but the average performance is the second best which is 3.1 lower than the best one, that is, Haar-like + HSV. Therefore, Haar-like + HSV is selected as feature fusion strategy for our following experiments.

Candidate samples selection. In this section, we need to select a small number of candidate samples to narrow the scope of the target searching. These candidate samples are obtained by the composite similarity weights, where the optimal similarity weight values need to be determined. In this section, we discuss the influence of the maximum value of synthetic weights WN on the tracking effect. The experimental results are shown in Table 2.

In order to ensure the rationality of the experiment, we do not adopt the dictionary update strategy here. Based on the above experimental data, we can obtain the curve of CLE versus WN (Figure 5).

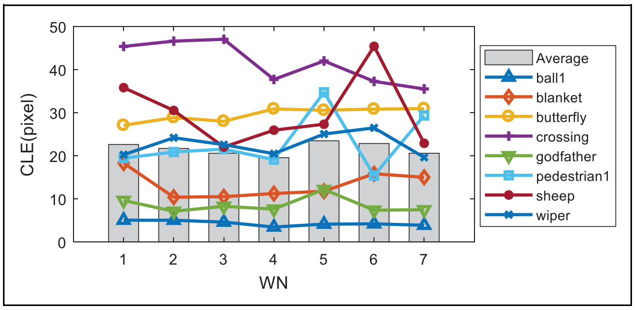

Figure 5. The average CLE variation for eight sequences. CLE: center location error.

Table 3. The distribution of 11 challenging attributes in the evaluating sequence set: IV, SV, OCC, DEF, MB, IPR, OPR, OV, BC, LR, and FM.

	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
Frequency	32	47	49	42	37	26	32	34	11	24	8

IV: illumination variation; SV: scale variation; OCC: partial or full occlusion; DEF: non-rigid object shape deformation; MB: motion blur; IPR: in-plane rotation; OPR: out-of-plane rotation; OV: out of view; BC: background clutters; LR: low resolution; FM: fast motion.

In Figure 5, the broken line indicates the change in the effect of a single video tracking. The histogram shows the average tracking effect of all videos. As shown in Figure 5, the average value is significantly increased when WN is greater than 4 and the tracking results of some sequences are also significantly changed, such as *pedestrian1*, *sheep*, and so on. In the method evaluation experiment, we set WN to 3 in the experiment.

Experimental evaluation

In the performance evaluation section, we mainly compare the proposed method against eight state-of-the-art methods including adaptive local sparse appearance model-based tracker (ASLA[1]), incremental learning-based tracker (IVT[2]), L1 sparse tracker using APG (L1APG[3]), compress tracker (CT[6]), context tracker (CXT[7]), online robust image alignment tracker (ORIA[9]), online boosting tracker (OAB[8]), and tracking learning-detection tracker (TLD[10]). The qualitative and quantitative experimental results are carried out with a useful analysis. All 74 RGB sequences on OTB100[37] are used as evaluating sequence set, and the distribution of all challenging attributes in the evaluating sequence set is shown in Table 3.

Quantitative analysis. In this section, the tracking results based on precision plots and success plots are used to comprehensively evaluate the performance of different methods on OTB100.[37] The legend of precision plots shows the values at the error threshold of 20 pixels, and the legend of success plots show the area under curve (AUC) values. The overlap score is a measure of the overlap range of the tracking result and the ground truth tracking box, defined as $OS = intersection\ area/union\ area$, where intersection area and union area are the intersection and union of two regions, respectively.

Figure 6 shows the overall tracking precision plots and success plots of all nine methods on 74 RGB sequences of OTB100.[37] The precision score and success score of our approach are ranked first, higher than the second methods by 10.6% and 7.4%, respectively. As can be seen from the precision plots of one-pass evaluation (OPE), as the location error threshold increases, the precision of other trackers grows slowly, and our algorithm improves a lot. In the precision plots of OPE, the success score of our method is significantly higher than the other methods.

Table 4 shows the performance of our method and eight benchmarking methods in terms of success plots and AUC scores on different attributes. The average AUC value of our method, TLD and CXT trackers are top 3 on 11 attributes. TLD and CXT trackers perform well on attributes of fast motion (FM), motion blur (MB), out of view (OV), and low resolution (LR) due to dense sampling. ASLA tracker performs better on occlusion (OCC), scale variation (SV), and non-rigid object shape deformation (DEF) attributes by its local representation.

Figure 7 shows the ranking of success plots of all benchmarking methods on the 11 challenging attributes. On challenging attributes of SV, OCC, out-of-plane rotation (OPR), DEF, FM, MB), OV, in-plane rotation (IPR), and LR, the success plot of our method ranks the first. Despite the lack of a scale-changing mechanism, our method still has the best performance with 0.390 score on SV attribute. In similar methods, ASLA using local information also has a good score on SV attribute, but its score is lower than our method by 3%.

The AUC scores of our method are higher than the second method ASLA by 5.8% and 7.4% on the attributes OPR and DEF, respectively, which shows the effectiveness of our feature selection mechanism in the target appearance change. ASLA and TLD trackers use local information and have good scores on OCC attribute, which are 5.1% lower than our method. On attributes FM and MB, our method is 8.3% and 7.5% higher than the second method CXT, respectively. The ASLA tracker used local information and had the best results on background clutter (BC) and illumination variation (IV) attributes, and the success rate score of ASLA is 0.397 which is better than other similar methods.

Qualitative analysis. Figure 8 shows the tracking process of eight similar trackers and our method in the several RGB sequences. In Figure 8, our method has good tracking performance on the attribute of MB and FM. In sequences *Deer* and *BlurOwl*, although tracking drift sometimes

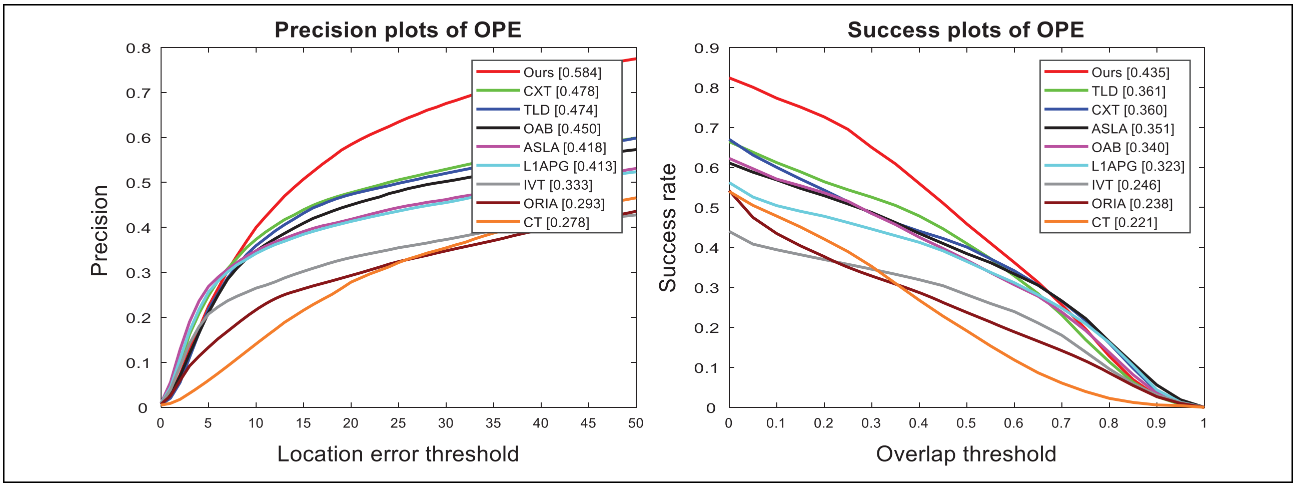

Figure 6. The comprehensive precision plots (left) and success plots (right) of comparison methods on 74 RGB sequences of OTB100.[37]

Table 4. The AUC value of all trackers in different attributes.[a]

	Ours	ASLA	IVT	OAB	L1APG	TLD	CT	ORIA	CXT
IV	**0.357**	**0.387**	0.263	0.278	0.324	0.343	0.215	0.268	**0.344**
OPR	**0.406**	**0.348**	0.232	0.290	0.264	**0.342**	0.236	0.252	0.328
SV	**0.390**	**0.360**	0.241	0.308	0.287	**0.348**	0.225	0.247	0.345
OCC	**0.408**	**0.357**	0.265	0.299	0.295	**0.357**	0.208	0.263	0.311
DEF	**0.387**	**0.313**	0.171	0.252	0.253	**0.292**	0.204	0.170	0.240
MB	**0.472**	0.213	0.188	0.363	0.322	**0.366**	0.180	0.171	**0.397**
FM	**0.452**	0.218	0.165	**0.366**	0.298	0.358	0.183	0.162	**0.369**
IPR	**0.409**	**0.356**	0.222	0.311	0.290	0.343	0.241	0.263	**0.373**
OV	**0.338**	0.264	0.190	0.217	0.201	**0.325**	0.185	0.153	**0.328**
BC	**0.380**	**0.397**	0.225	0.261	0.291	0.271	0.252	0.185	**0.293**
LR	**0.350**	0.325	0.274	0.301	0.334	**0.342**	0.208	0.229	**0.345**
Average	**0.395**	0.322	0.221	0.295	0.287	**0.335**	0.212	0.215	**0.334**

AUC: area under curve; ASLA: adaptive local sparse appearance model-based tracker; IVT: incremental learning-based tracker; OAB: online boosting tracker; L1APG: L1 sparse tracker using APG; TLD: tracking learning-detection tracker; CT: compress tracker; ORIA: online robust image alignment tracker; CXT: context tracker; IV: illumination variation; OPR: out-of-plane rotation; SV: scale variation; OCC: partial or full occlusion; DEF: non-rigid object shape deformation; MB: motion blur; FM: fast motion; IPR: in-plane rotation; OV: out of view; BC: background clutters; LR: low resolution.

[a] Bold data indicate the AUC scores are top three.

occurs, our approach can readjust the tracking position through the positive and negative templates when drifting is not severe. In general, OAB has a good tracking effect in these two videos, but it is prone to have tracking drift problems when the target moves fast, as shown by #0025 of *Deer* and #0390 of *BlurOwl*. CXT tracker can well recognize the target information in these sequences, but when the target blur and FM occur, the scale of the tracking will be abnormal.

Sequences *bolt*, *bolt2*, and *basketball* are typical of the target DEF. CT tracker is a tracking method based on compressed sensing and has good performance in target DEF, as shown in sequence *bolt2*. However, it appears that many tracking failures occur in sequences *deer* and *BlurOwl*, which indicates that CT tracker suffers from target FM easily. Our method performs well in the challenges of target DEF, but not in IV. As shown in #0700 of sequence *basketball*, our method shows significant tracking drift when there is a noticeable illumination change.

In sequence *David3*, most trackers suffer from OCC and BC, but our method can effectively deal with short-term occlusion of a large area because the adaptive dictionary update strategy minimizes occlusion interference. From the sequences *David3* and *couple* in Figure 6, we can see that OAB, CT, and the proposed approach have good tracking performance in the background changes. TLD has the problems of tracking drift and target lost. Both ORIA and CXT trackers are affected by small-range occlusion, which causes to tracking failure. In sequence *David3* #0146, a wide range of occlusions also leads to tracking failures of CT and OAB. Our method effectively identifies the target location in these cases and does a good job for the rest of tracking tasks.

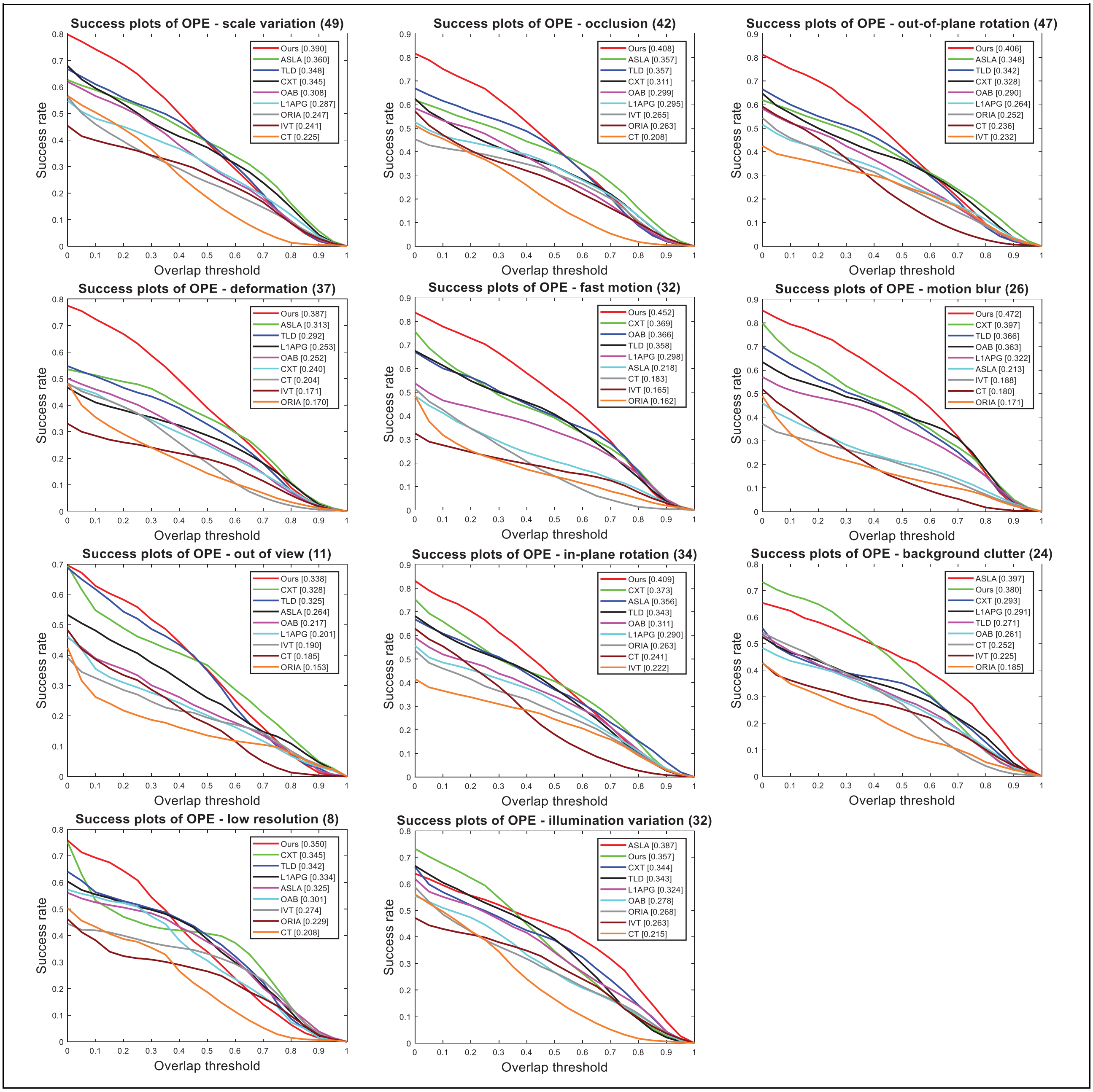

Figure 7. Success plots of 11 challenging attributes on all 74 RGB sequences of OTB100.[37]

In the last two sequences, *Lemming* and *DragonBaby* contain multiple challenge attributes such as SV, OCC, rotation (IPR or OPR), and OV. It can be seen in Figure 6 that the tracking drift is easily occurred when the target fast rotation, SV, and BC occur simultaneously. In #1010 and #1078 of sequence *Lemming*, TLD, CXT, and CT trackers have obvious tracking drift due to fast rotation, while OAB and our method do not suffer from that and perform better results. In #0084 and #0096 of sequence *DragonBaby*, our method performed well for target fast rotation and background interference. CXT tracker has tracking scale anomalies, and other methods have repeatedly experienced tracking drift and tracking failure. In sequence *human8* #0054 and #0070, most trackers have tracking failures when both illumination and scale changes occur. At the #0101 and #0126 frames of sequence *human8*, the true scale of the target is significantly smaller, and the result area selected by our method contains a large amount of background information. This situation makes the performance of our tracker unstable and prone to tracking failure.

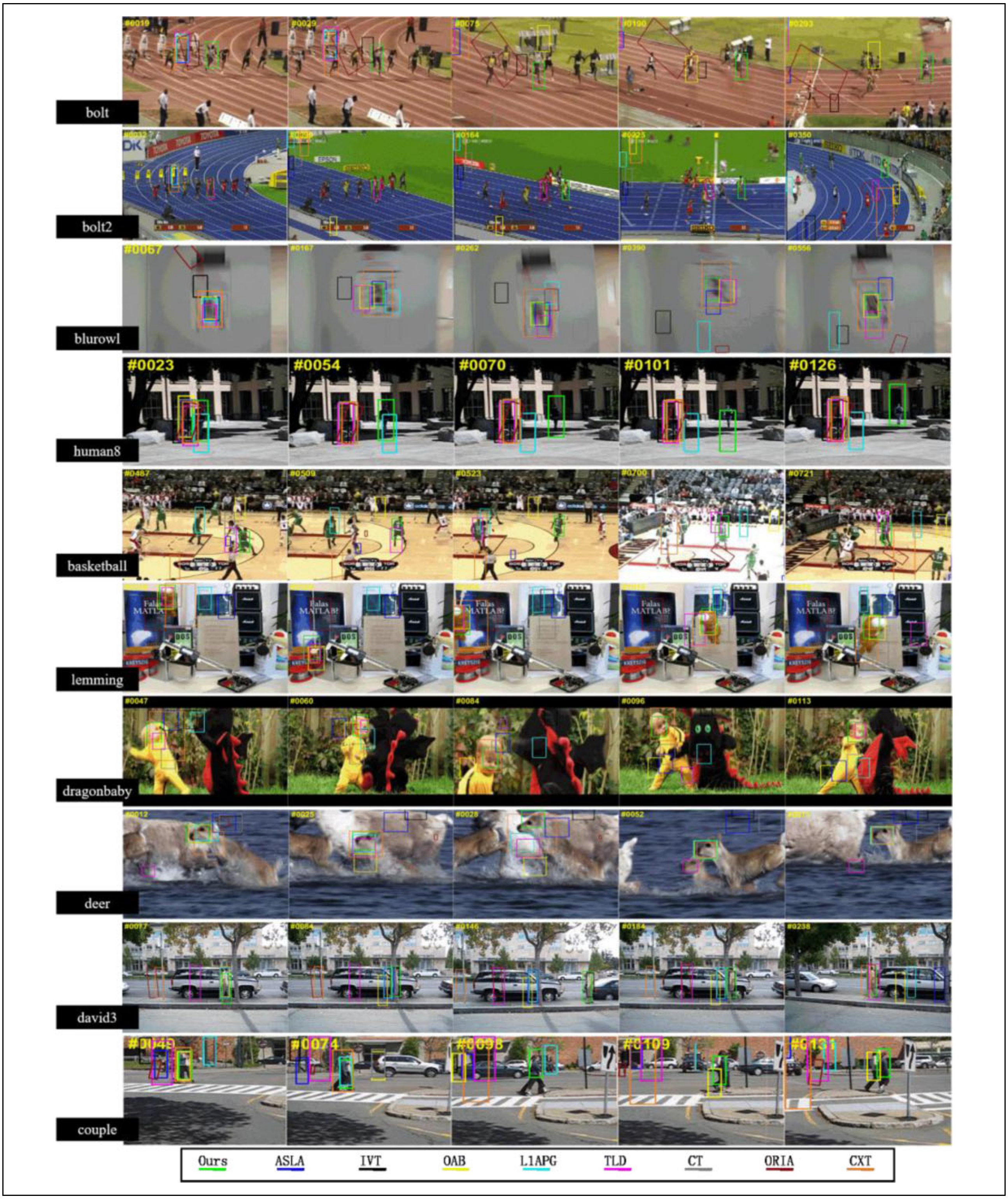

Figure 8. Comparison of the proposed approach with eight benchmarking methods ASLA, IVT, OAB, L1APG, TLD, CT, ORIA, and CXT. ASLA: adaptive local sparse appearance model-based tracker; IVT: incremental learning-based tracker; OAB: online boosting tracker; L1APG: L1 sparse tracker using APG; TLD: tracking learning-detection tracker; CT: compress tracker; ORIA: online robust image alignment tracker; CXT: context tracker.

Conclusion and future work

This article proposes a novel visual tracking method based on the weighted discriminative dictionaries and a pyramidal feature selection strategy. We utilize color features and noncolor features of the training samples to build multiple discriminate dictionaries. Then, we use the position information of samples to assign weights to the base vectors in dictionaries. These weights are used to optimize the process of target searching for selection of candidate samples, so that the frequency of abnormal samples can be effectively reduced. In the tracking process, for reducing the introduction of interference information in the dictionary and improving the tracking efficiency, we gradually update the dictionary based on noise analysis of the sparse coefficients. During the incremental update process, we sample the pool to maintain the appearance change of the target and obtain the current foreground and background information. The positive sample pool also uses a random replacement maintenance strategy to maintain the class balance of the samples. Experimental results on the all RGB sequences on OTB100[37] show that the proposed method is effective to deformation, occlusion, and other challenges in object tracking.

We will further investigate this work. First, in the video scene with cluttered background, the target is easy to be misjudged. We plan to increase the fusion of three or more features to enhance the accuracy of the target representation. Secondly, when the target scale changes, it is easy to drift away even though there are different scales of sampling. So, the mechanism of dealing with the change of target scale should be further studied.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China (grant numbers 61772144 and 61672008), Guangdong Provincial Application-oriented Technical Research and Development Special Fund Project (grant number 2016B010127006), Foreign Science and Technology Cooperation Plan Project of Guangzhou Science Technology and Innovation Commission (grant number 201807010059), the Scientific and Technological Projects of Guangdong Province (grant number 2017A050501039), Innovation Team Project (Natural Science) of the Education Department of Guangdong Province (grant number 2017KCXTD021), Innovation Research Project (Natural Science) of Education Department of Guangdong Province (grant number 2016KTSCX077), and Foundation for Youth Innovation Talents in Higher Education of Guangdong Province (grant number 2018KQNCX139).

ORCID iD

Penggen Zheng https://orcid.org/0000-0001-6225-7002

References

1. Lu H, Jia X, and Yang MH. Visual tracking via adaptive structural local sparse appearance model. In: *Proceeding computer vision and pattern recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012. pp. 1822–1829. IEEE.
2. Ross DA, Lim J, Lin RS, et al. Incremental learning for robust visual tracking. *Int J Comput Vision* 2008; 77(1–3): 125–141.
3. Bao C, Wu Y, Ling H, et al. Real time robust L1 tracker using accelerated proximal gradient approach. In: *Proceeding computer vision and pattern recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012, pp. 1830–1837.
4. Henriques JF, Rui C, Martins P, et al. High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 2014; 37(3): 583–596.
5. Mei X and Ling H. Robust visual tracking using ℓ 1 minimization. In: *IEEE international conference on computer vision*, Kyoto, Japan, 29 July 2010. IEEE.
6. Zhang K, Zhang L, Yang MH, et al. Real-time compressive tracking. In: *Computer vision (ECCV)*, Florence, Italy, 7–13 October 2012, pp. 864–877.
7. Dinh TB, Vo N, and Medioni G. Context tracker: exploring supporters and distracters in unconstrained environments. In: *Proceeding computer vision and pattern recognition (CVPR)*, Colorado Springs, CO, USA, 20–25 June, 2011, pp. 1177–1184. IEEE.
8. Grabner H and Bischof H. On-line boosting and vision. In: *Proceeding computer vision and pattern recognition (CVPR)*, New York, NY, USA, 17–22 June 2006, pp. 260–267.
9. Ling H. Online robust image alignment via iterative convex optimization. In: *Proceeding computer vision and pattern recognition (CVPR)*. Providence, RI, USA, 16–21 June 2012, pp. 1808–1814. IEEE.
10. Kalal Z, Mikolajczyk K, and Matas J. Tracking-learning-detection. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(7): 1409–1422.
11. Wang N, Li S, Gupta A, et al. Transferring rich feature hierarchies for robust visual tracking. In: *Computer science*, Ithaca, NY, USA, 19 January 2015, vol. 1.
12. Qi Y, Zhang S, Qin L, et al. Hedging deep features for visual tracking. *IEEE Trans Pattern Anal Mach Intell* 2019; 41(5): 1116–1130.
13. Qi Y, Zhang S, Zhang W, et al. Learning attribute-specific representations for visual tracking. In: *Thirty-third AAAI conference on artificial intelligence (AAAI)*, Honolulu, Hawaii, USA, 27 January–1 February 2019.
14. Zhang S, Qi Y, Jiang F, et al. Point-to-set distance metric learning on deep representations for visual tracking. *IEEE Trans Intell Transport Syst* 2018; 19(1): 187–198.
15. Qi Y, Zhang S, Qin L, et al. Hedged deep tracking. In: *IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 27–30 June 2016, pp. 4303–4311.

16. Zhong B, Bai B, Li J, et al. Hierarchical tracking by reinforcement learning based searching and coarse-to-fine verifying. *IEEE Trans Image Process* 2019; 28(5): 2331–2341.
17. Zhou Q, Zhong B, Zhang Y, et al. Deep alignment network based multi-person tracking with occlusion and motion reasoning. *IEEE Trans Multimedia* 2019; 21: 1183–1194.
18. Jiang Z, Lin Z, and Davis LS. Label consistent K-SVD: learning a discriminative dictionary for recognition. *IEEE Trans Pattern Anal Mach Intell* 2013; 35(11): 2651–2664.
19. Jin Z, Su Z, Wu H, et al. Robust tracking via discriminative sparse feature selection. *Visual Comput* 2015; 31(5): 575–588.
20. Wang N and Yeung DY.Learning a deep compact image representation for visual tracking. In: *Advances in neural information processing systems*, Lake Tahoe, Nevada, USA, 5–10 December 2013, pp. 809–817.
21. Song Y, Ma C, Wu X, et al. Vital: visual tracking via adversarial learning. In*: Conference on computer vision and pattern recognition*, Salt Lake City, UT, USA, 18–22 June 2018, pp. 8990–8999.
22. Guo Q, Feng W, Zhou C, et al. Learning dynamic Siamese network for visual object tracking. In: *IEEE international conference on computer vision (ICCV)*, Venice, Italy, 22–29 October 2017, pp. 1781–1789. IEEE Computer Society.
23. Fan H and Ling H. SANet: structure-aware network for visual tracking. In: *Computer vision and pattern recognition workshops*, Honolulu, HI, USA, 21–26 July 2017, pp. 2217–2224. IEEE.
24. Wang Z, Ren J, Zhang D, et al. A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos. *Neurocomputing* 2018; 287: 68–83.
25. Lan X, Ma AJ, Yuen PC, et al. Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. *IEEE Trans Image Process* 2015; 24(12): 5826–5841.
26. Lu H, Jia X, and Yang MH. Visual tracking via adaptive structural local sparse appearance model. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012. IEEE Computer Society.
27. Han Z, Jiao J, Zhang B, et al. Visual object tracking via sample-based adaptive sparse representation (AdaSR). *Pattern Recogn* 2011; 44(9): 2170–2183.
28. Qi Y, Qin L, Zhang J, et al. Structure-aware local sparse coding for visual tracking. *IEEE Trans Image Process* 2018; 27(8): 3857–3869.
29. Li Z, Zhang J, Zhang K, et al. Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning. *IEEE Trans Image Process* 2018; 27(9): 4478–4489.
30. Zhang S, Yao H, Sun X, et al. Sparse coding based visual tracking: review and experimental comparison. *Pattern Recogn* 2013; 46(7): 1772–1788.
31. Hong Z, Mei X, Prokhorov D, et al. Tracking via robust multi-task multi-view joint sparse representation. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, NSW, Australia, 1–8 December 2013, pp. 649–656.
32. Zhang S, Zhou H, Jiang F, et al. Robust visual tracking using structurally random projection and weighted least squares. *IEEE Trans Circ Syst Vid Tech* 2015; 25(11): 1749–1760.
33. Zhang S, Lan X, Yao H, et al. A biologically inspired appearance model for robust visual tracking. *IEEE Trans Neural Network Learn Syst* 2017; 28(10): 2357–2370.
34. Yan Y, Ren J, Sun G, et al. Unsupervised image saliency detection with Gestalt-laws guided optimization and visual attention based refinement. *Pattern Recogn* 2018; 79: 65–78.
35. Yan Y, Ren J, Zhao H, et al. Cognitive fusion of thermal and visible imagery for effective detection and tracking of pedestrians in videos. *Cogn Comput* 2017; 10(1): 94–104.
36. Kristan M, Eldesokey A, Xing Y, et al. The visual object tracking VOT2017 challenge results. In: *IEEE international conference on computer vision workshop*, Venice, Italy, 22–29 October 2017, pp. 1949–1972. IEEE Computer Society.
37. Wu Y, Lim J, and Yang MH. Object tracking benchmark. *IEEE Trans Pattern Anal Mach Intell* 2015; 37(9): 1834–1848.
38. Lienhart R and Maydt J. An extended set of haar-like features for rapid object detection. In: *International conference on image processing*, Rochester, NY, USA, 22–25 September 2002, *vol. 1,* pp. I-900–I-903. IEEE.