Robust object tracking based on weighted subspace reconstruction error with forward-backward tracking criterion

Tao Zhou,[a,b] Kai Xie,[a,b] Junhao Zhang,[a,b] Jie Yang*[a,b], Xiangjian He[c]
[a]Shanghai Jiao Tong University, Institute of Image Processing & Pattern Recognition, Department of Automation, 800 Dongchuan Road, Shanghai, P.R.China, 200240
[b]Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai
[c]University of Technology, Sydney (UTS), School of Computing and Communications, NSW 2007, Australia

Abstract. It is a challenging task to develop an effective and robust object tracking method due to factors such as severe occlusion, background clutters, abrupt motion, illumination variation and so on. In this paper, a novel tracking algorithm based on weighted subspace reconstruction error is proposed. The discriminative weights are defined through minimizing reconstruction error with positive dictionary while maximizing reconstruction error with negative dictionary. Then, confidence map for candidates is computed through subspace reconstruction error. Finally, the location of the target object is estimated by maximizing the decision map which is combined discriminative weights and subspace reconstruction error. Furthermore, the new evaluation method based on forward-backward tracking criterion to verify the robustness of the current tracking performance in updating stage, which can reduce the accumulated error effectively. Experimental results on some challenging video sequences show that the proposed algorithm performs favorably against eleven state-of-the-art methods in terms of accuracy and robustness.

Keywords: Object tracking; subspace reconstruction; discriminative weights; forward-backward tracking criterion.

Address all correspondence to: Jie Yang, E-mail: jieyang@sjtu.edu.cn

1 Introduction

Object tracking is one of the most research topics due to its wide range of applications such as behavior analysis, activity recognition, video surveillance, and human-computer interaction. Although it has obtained a significant progress in the past decades, developing an efficient and robust tracking algorithm is still a challenging task due to numerous factors such as illumination variation, partial occlusion, pose change, abrupt motion, background clutter and so on.

The main tracking algorithms can be classified into two kinds: generative[1–7] or discriminative methods.[8–12]

Generative methods focus on searching for the regions which are the most similar to the tracked targets with minimal reconstruction errors of tracking. Adaptive models including the WSL tracker[13] and IVT method[4] have been proposed to handle appearance variation. Adam et.al[1] used several

fragments to build an appearance model to handle partial occlusion and pose variation. Recently, sparse representation methods have been used to represent an object by a set of trivial target templates and trivial templates[3,14] to deal with partial occlusion, pose variation and so on. Thus, it is critical to construct an effective appearance model in order to handle various challenging factors. Furthermore, generative methods discard useful information surrounding target regions that can be exploited to better separate objects from backgrounds.

Discriminative methods treat tracking as a classification problem that distinguishes the tracked targets from the surrounding backgrounds. A tracking technique called tracking by detection has been shown to have promising results in real-time. This approach trains a discriminative classifier online to separate an object from its background. Collins et al.[15] selected discriminative features online to improve the tracking performance. Boosting method has been used for object tracking through combining weak classifiers to establish a strong classifier to select discriminative features, and some online boosting feature selection methods have been proposed for object tracking.[16,17] Babenko et al.[8] proposed a novel online MIL algorithm for object tracking that achieves superior results with real-time performance. An efficient tracking algorithm based on compressive sensing theories was proposed by Zhang et al.[9] It uses low dimensional features randomly extracted from high dimensional multi-scale image features in the foreground and background, and it achieves better tracking performance than other methods in terms of robustness and speed.

The above tracking methods have shown promising performance. However, they have some shortcomings. Firstly, although the goal of a generative method is to learn an object appearance model, an effective searching algorithm and measuring method to match candidate samples to an object model are difficult to obtain. Secondly, background varies broadly during a tracking process, so it is difficult to achieve the aim of a discriminative method to distinguish a target region from

a complicated background when the target looks similar to its background. Therefore, it is very difficult to construct a discriminative target model.

2 Motivation

Subspace representation is possibly the most common choice for appearance models in object tracking, mainly because it is easy to compute and robust for scale variation, rotation, pose changes and illumination variation.[18] Ross et al.[4] proposed the IVT method which represents the tracked target by a low dimensional PCA subspace and assumes that the error is Gaussian distributed with small variances. Hence, the representation coefficient can be obtained by a simple projection transformation. Furthermore, it is effective to handle appearance change caused by illumination variation. However, it has following drawbacks. Firstly, ordinary least squares methods have been shown to be sensitive to occlusion and background clutter based on reconstruction error. Secondly, the update scheme uses new observations to update the subspace model without detecting outliers and processing them accordingly, so it will cause inaccurate update for the subspace of the target to bring tracking drifts and big tracking accumulated errors.

Recently, sparse representation has been introduced to the tracking task.[3,19–23] Mei et.al proposed the L1 tracking method.[3] For tracking in their algorithm, a candidate sample can be sparsely represented by a template set or dictionary, and its corresponding likelihood is determined by the reconstruction error with respect to target templates. The L1 tracker has obtained promising robustness compared with many existing trackers. However, the dictionary can not consider the background, while the tracker uses an over-complete dictionary (an identity matrix) to represent the background and noises. As a result, it may not discriminate the objects against complicated background.

Wang et al. proposed an online algorithm based on local sparse representation for robust object tracking.[19] It uses the sparse codes of local image patches with an over-complete dictionary for object representation, and trains a linear classifier to separate the target object from the background. However, the linear classifier is not robust to background clutters. Inspired by using generative and discriminative models together to enhance the robustness of the tracker, a structured collaborative representation-based visual tracking algorithm is proposed.[20] Firstly, positive and negative samples are represented by their structured collaborative representation coefficients which obtained by encoding sparse representation with target and background templates, then the structured collaborative representation coefficients are used to train a Bayes classifier which can offer each candidate a classification score. This method is similar to Wang's work,[19] as sparse coefficients are used for object representation, and then a classifier is trained to distinguish target from background. Liu et al. proposed a fast object tracking method with two stage sparse optimization.[23] However, the tracker essentially depended on on online self-training classifier and it is susceptible to drifting. A robust visual tracker based on structured sparse representation appearance model proposed in,[14] it adds the structured information without utilizing the background template. Hong et al. proposed a robust multi-task multi-view joint sparse learning method for visual tracking based on particle filter framework.[21] The method can exploit the underlying relationship shared by different views and different particles, but also it can capture the frequently emerging outlier tasks.

There are three drawbacks of some existing methods based sparse representation as follows. Firstly, Sparse representation coefficients are used for object representation and the tracking task is treated as a binary classification problem. The trained linear classifier is sensitive to background clutter and appearance change. Secondly, the prototype L1 tracker is vulnerable to failure in the case of the dictionary is updated with background image patches or inaccurate tracked results.

This is because the wrong templates are also possibly activated for approximating the observations and achieve high likelihood with the background image patches or inaccurate tracked results. Thirdly, some tracking methods are only trained based on object appearance without utilizing the information from the background, which does not ensure distinguish ability.

To overcome these flaws mentioned above, a novel tracking algorithm via weighted subspace reconstruction error is proposed in this paper. As shown in Fig.1, we firstly define the discriminative weights based on sparse construction using the positive dictionary and negative dictionary respectively. It is similar to Fisher linear discriminant criterion, the goal of discriminative weights is to minimize reconstruction error using positive dictionary while maximizing reconstruction error using negative dictionary. The discriminative weights can reduce the sensitiveness of a failure in the case of the dictionary is updated with background image patches, as the background image patches can not be used for updating the dictionary because of their low of discriminative weights. Secondly, the confidence map for candidates is computed through subspace reconstruction error. In the last step, the optimal location is estimated by maximizing the decision map which combines discriminative weights and subspace reconstruction error. In updating stage, a forward-backward tracking criterion to verify the robustness of the current tracking performance. The evaluating criterion can effectively handle tracking outliers and reduce the cumulative errors. The details of our method are shown in Fig.1. Empirical results on some challenging video sequences demonstrate the superior performance of our method in terms of accuracy and robustness to state-of-the-art tracking methods.

The main contributions of this paper are as follows.

1. The discriminative weights are defined to distinguish the target from complex background

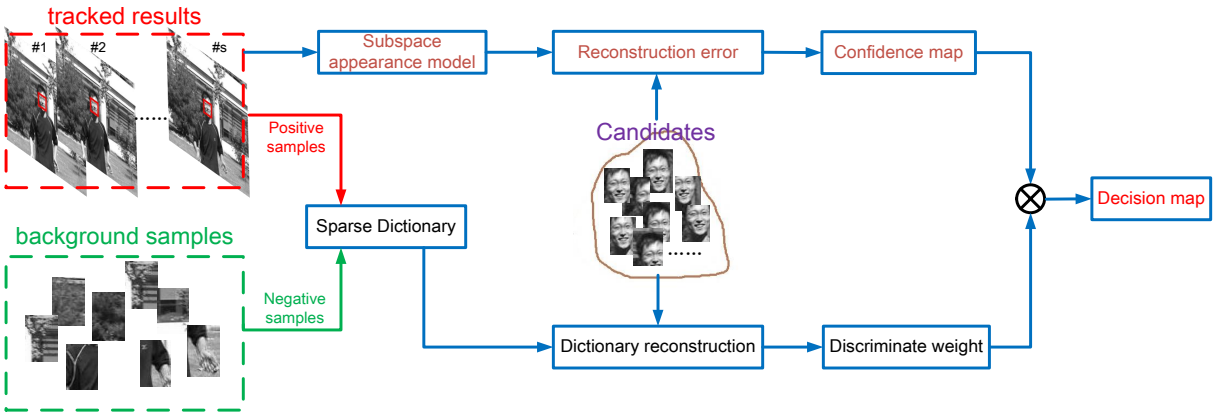


Fig 1 The flow of our proposed tracking algorithm.

clutter accurately. The use of discriminative weights is to minimize the reconstruction error using a positive dictionary while maximizing the reconstruction error using a negative dictionary. The discriminative weights are obtained by the positive and negative dictionary reconstructions respectively and can ensure the distinguish ability.

2. The forward-backward tracking criterion is used to evaluate the current tracking performance, which can be adopted to decide whether to update the subspace appearance model and reduce the accumulated errors effectively.

3. The decision map combining discriminative weights and subspace reconstruction error can make use of the advantages of sparse representation and subspace appearance model, which can enhance the robustness to multiple challenging factors.

4. Experimental results on some challenging video sequences show that the proposed algorithm outperforms twelve state-of-the-art methods in terms of accuracy and robustness.

This is is an extension of our paper showing preliminary results in.[24] The rest of this paper is organized as follows. Details of our proposed method based on weighted subspace reconstruction

error are demonstrated in Section 3. Experimental results are shown and analyzed in Section 4. The conclusion is presented in Section 5.

3 Proposed method

3.1 Particle filter based tracking formulation

In our method, we estimate the target states using the Bayesian inference framework. Supposed that the observations of the target $Z_1 : t = \{Z_1, Z_2, \cdots, Z_t\}$ up to time t, the target state x_t can be computed by the maximum a posteriori (MAP) estimation as follows:

$$x_t = \underset{x_t}{\operatorname{argmax}}\, p(x_t|Z_{1:t}) \tag{1}$$

In our method, we estimate the target states using the Bayesian inference framework, the target state x_t can be computed by the maximum a posteriori (MAP) estimation as follows:

$$\hat{x_t} = \underset{x_t}{\operatorname{argmax}}\, p(x_t|z_{1:t}) \tag{2}$$

The posteriori probability $p(x_t|z_{1:t})$ can be inferred by Bayesian theory

$$p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|z_{1:t-1}) \tag{3}$$

with

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{4}$$

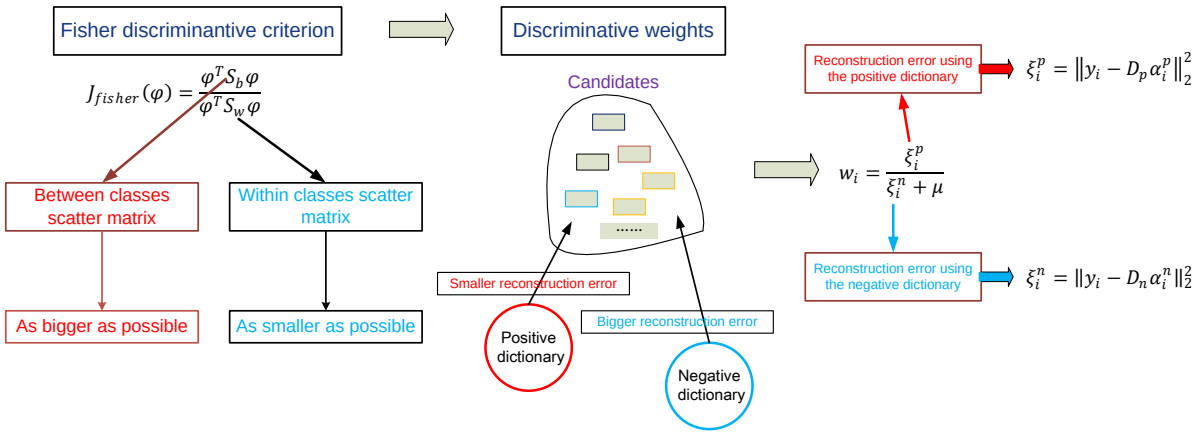


Fig 2 Illustration of details to define the discriminative weights. It is similar to the linear discriminant criterion: the goal is to minimize the reconstruction error using a positive dictionary while maximizing the reconstruction error using a negative dictionary. The discriminative weights are obtained by the positive and negative dictionary reconstructions respectively and can ensure the distinguish ability.

In the particle filter framework, the posterior $p(x_t|z_{1:t})$ can be computed approximately by a finite set of random sampling particles. In the proposed method, the target state x_t is modeled by a six-dimensional parameter vector for affine transformation. We model each transformation parameter independently by a Guassion distribution between two consecutive frames. The observation model $p(z_t|x_t)$ reflects the similarity between the target template and a candidate.

In our method, $p(z_t|x_t)$ is formulated to minimize the weighted subspace reconstruction error, which is defined by

$$p(z_t|x_t) \propto e^{-w*\varepsilon} \tag{5}$$

where w and ε are detailed in following sections.

3.2 Discriminative weights

We define the discriminative weights based on sparse construction using the positive dictionary and negative dictionary respectively. It is similar to Fisher linear discriminant criterion, the goal

of discriminative weights is to minimize the reconstruction error using a positive dictionary while maximizing the reconstruction error using a negative dictionary. The positive and negative samples can ensure the distinguish ability. The details of discriminative weights are shown in Fig.2.

In our method, tracked results are used for the construction of the positive dictionary and some samples that are away from the target are used for negative dictionary. In this way, we can obtain a dictionary consisting of positive and negative samples as follows. Firstly, we assume the location in the first s frames have been obtained by nearest matching method. Tracked results are collected to form the positive dictionary $D_p = \{D_1^p, D_2^p, \cdots, D_i^p\}, i = 1, 2, \cdots, N_p$. Then, we sample some image patches away from the current location of the target to establish the negative dictionary $D_n = \{D_1^n, D_2^n, \cdots, D_i^n\}, i = 1, 2, \cdots, N_n$. The final dictionary is represented as $D = [D_p, D_n] \in \Re^{d \times (N_p + N_n)}$, where N_p and N_n are the numbers of positive samples and negative samples, respectively. Each column in dictionary D is obtained through L2 normalization on the vectorized positive and negative dictionary.

With the sparsity assumption, the candidates within the target region can be represented as the linear combination with only a few basis elements of the dictionary by solving

$$\underset{\alpha_i}{\operatorname{argmin}} \|y_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \tag{6}$$

where α denotes the corresponding sparse code of each candidate, and λ_i is control parameter. Similar to the Fisher linear discriminant analysis,[25] the tracking is regarded as a process to find a classifier given a target and its background. The aim of object tracking is to find the candidate which should produce a smaller reconstruction error using the positive dictionary, but vice versa

using the negative dictionary. The discriminative weights are defined as follows.

$$w_i = \frac{\xi_i^p}{\xi_i^n + \mu} \tag{7}$$

To normalize

$$w_i = \frac{w_i}{\sum w_i} \tag{8}$$

where $\xi_i^p = \|y_i - D_p\alpha_i^p\|_2^2$ and $\xi_i^n = \|y_i - D_n\alpha_i^n\|_2^2$, which denote the reconstruction errors using two different sub-dictionaries, and μ is a constraint factor to avoid non-division. From the above equation, a candidate having a smaller discriminative weight is more likely to be the target, and vice versa. The discriminative weight can effectively distinct a target object from a complicated background. More importantly, it reflects the possibility of an object being a target by encoding the sparse coefficients using positive and negative dictionaries.

3.3 Subspace reconstruction error

A candidate with a smaller reconstruction error based on the subspace representation is more likely to be the target. Based on this concern, the reconstruction error of each candidate is computed based on the subspace model generated from a target template D_p using Incremental Principal Component Analysis (IPCA).

The eigenvectors form the normalized covariance matrix of template D_p, $U = [u_1, u_2, \cdots, u_l]$, which is corresponding to the largest l eigenvalues computed by PCA. Based on U , we can obtain

projection coefficient for each candidate by

$$\beta_i = U^T(y_i - \bar{y}) \tag{9}$$

where y_i denotes the candidate sample, and $\bar{y}$ is the mean feature of template D_p. Then, the subspace reconstruction error of each candidate is computed by

$$\varepsilon_i = \|y_i - (U\beta_i + \bar{y})\|^2 \tag{10}$$

where ε_i indicates a candidate is more likely to be a target object with a smaller reconstruction error. We gradually learn a low-dimensional subspace representation, which can adapt the on-line target appearance change.

3.4 Decision map

The smaller discriminative weight and reconstruction error one candidate has, the closer it is to the real location in the coming frame. Therefore, we use maximum posterior to estimate observation model $p(z_t|x_t)$. In our tracking algorithm, the final decision map is defined by

$$p_i = e^{-w_i * \varepsilon_i} \tag{11}$$

and the optimal state x_t at frame t is estimated by

$$\hat{x_t} = \underset{i}{\operatorname{argmax}}\, p_i \tag{12}$$

where w_i represents the discriminative weight of each candidate sample, and ε_i represents the subspace construction error. The tracking result is one candidate has the highest confidence value.

3.5 Update scheme

For the dictionary $D = [D_p, D_n]$, we update the negative dictionary every 5 frames to sample away from the current tracking result. To update the positive dictionary, the sample on the current tracked location is added and then to delete the oldest sample in the positive dictionary .

To construct a subspace appearance model, if we directly update the template with new observations, errors are likely to be accumulated and the tracker will drift away from the target. To a robustness tracking algorithm, if the performance is good by implementing a tracking algorithm from the frame t to the frame $t+1$, it also obtains the better performance by implementing the tracker from the frame $t+1$ back to the frame t. Therefore, we use a forward and backward[26] tracking criterion to evaluate the current tracking performance. Fig.3 shows the flow of forward-backward tracking method.

Forward tracking: starting from the current frame t to the next frame $t+1$, let us denote the target location in the t-th frame by x_t^*, the location obtained using our method in frame $t+1$ by x_{t+1}^* (see the red dashed arrow in Fig.3).

Backward tracking: starting from the current location x_{t+1}^*, we obtain a backward tracking result $x_t^{'}$ in the previous frame t using our method (see the green dashed arrow in Fig.3).

If the target is correctly tracked, x_t^* should be equal to the $x_t^{'}$. Thus, forward-backward tracking error is defined as follows:

$$error = \|x_t^* - x_t^{'}\|^2 \tag{13}$$

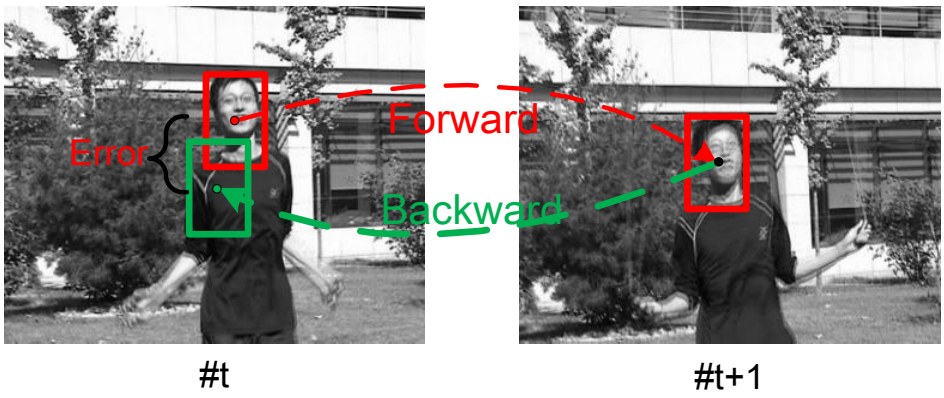


Fig 3 Illustration of evaluation criterion via forward and backward tracking method.

If the performance of one tracking method performs well, the error should be very small. However, the current tracking could drift away from the target when the error is very big, so we do not update the subspace model. A soft threshold is used to estimate the current tracking performance. If $error < \tau$ (τ is set to 5 in our experiments), we update the subspace appearance model, otherwise do nothing. The more details for appearance model updating are shown in the IVT method.[4]

3.6 Comparison with related work

It should be noted that the proposed tracking algorithm is significantly different from recently related methods including L1[3] and IVT[4] methods.

In L1 tracking method, a candidate sample can be sparsely represented by a template set or dictionary, and its corresponding likelihood is determined by the reconstruction error with respect to target templates. However, the algorithm selects only some samples around a target to model the target template and they do not consider background influence, while the tracker uses an overcomplete dictionary to represent the background and noises. As a result, it may not discriminate the objects against complicated background. In our method, we first select some samples away from the current object' location to model the negative dictionary, then the discriminative weight-

s are defined through minimizing reconstruction error with positive dictionary while maximizing reconstruction error with negative dictionary. The discriminative weights make use of the relationship between the object and background, so they helps our tracker to distinguish the object from complex background.

To update the subspace appearance model in IVT method, if we directly update the template with new observations, errors are likely to be accumulated and the tracker will drift away from the target. A forward-backward tracking criterion to evaluate the current tracking performance. Compared with directly updating appearance model, our method can handle tracking outliers and reduce the cumulative error.

4 Implementation and Experiments

4.1 Experimental setup

We evaluate the proposed tracking method based on weighted subspace reconstruction error using ten challenge video sequences with impact factors including abrupt motion, occlusion, illumination variation and background clutter (See Table 1). We compare our proposed tracker with other eleven state-of-the-art methods including: L1 tracker (L1),[3] real-time compressive tracking (CT),[9] multiple instance learning tracker (MIL),[8] incremental visual tracking (IVT),[4] fragment tracker (Frag),[1] weighted multiple instance learning tracker (WMIL),[11] MTT,[27] LSAT,[28] PN algorithm (PN),[29] VTD,[2] ODOT[19] and LOT.[30] For fair comparison, we adopt the source codes or binary codes provided by the authors with tuned parameters for best performance. For some trackers involving randomness, we repeat the experimental results 5 times on each sequence and obtain the averaged results.

Table 1 Evaluated video sequences.

Sequence	#Frames	Challenging Factors
Car4	659	illumination variation, scale change
Car11	393	illumination variation, scale change, background clutter
Jumping	313	abrupt motion
Caviar1	382	partial occlusion, scale change
Caviar2	500	partial occlusion, scale change
Caviar3	500	partial occlusion, scale change
Deer	71	abrupt motion, background clutter
Occlusion1	898	partial occlusion
Occlusion2	819	partial occlusion
DavidIndoor	462	illumination variation, scale change, out-plane rotation
Girl	501	partial occlusion, appearance change, rotation
Couple	140	partial occlusion, shaky, abrupt motion, background clutter

Table 2 Center location error (CLE). Red fonts indicate the best performance while the blue fonts indicate the second best ones. (Ours$^-$ represents our tracking method without the forward-backward tracking criterion).

Sequence	PN	L1	VTD	MIL	Frag	CT	WMIL	IVT	MTT	LSAT	LOT	ODOT	Ours$^-$	Ours
DavidIndoor	9.7	7.6	13.6	16.2	76.7	12.8	11.4	3.6	13.4	6.3	58.2	69.3	15.6	3.5
Occlusion1	17.7	6.5	11.1	32.3	5.6	19.5	23.5	9.2	14.1	5.3	21.3	6.7	6.2	5.2
Occlusion2	18.6	11.1	10.4	14.1	15.5	16.5	16.7	10.2	9.2	58.6	18.9	9.0	8.2	6.7
Caviar1	5.6	119.9	3.9	48.5	5.7	16.8	23.8	45.3	20.9	1.8	2.2	55.2	41.3	2.1
Caviar2	8.5	3.2	4.7	70.3	5.8	61.7	59.8	8.6	65.4	45.6	3.4	7.9	3.2	2.6
Caviar3	–	18.6	58.2	100.2	116.1	61.4	69.2	66.2	67.5	55.3	42.4	25.4	62.8	3.0
Car4	18.8	4.1	12.3	60.1	179.8	218.1	162.5	2.9	37.2	3.3	183.8	175.3	2.8	2.7
Car11	25.1	33.3	27.1	43.5	63.9	78.4	96.1	2.1	1.8	4.1	47.7	23.4	2.1	1.8
Deer	25.7	171.5	11.9	66.5	92.1	95.0	25.1	127.6	9.2	69.8	94.8	159.7	9.8	6.8
Jumping	3.6	92.4	62.7	9.9	58.5	47.4	64.4	36.8	19.2	55.2	6.2	13.9	22.9	5.5
Girl	23.2	62.4	21.4	32.2	18.0	38.6	44.2	48.4	23.9	143.3	16.1	12.3	18.6	13.5
Couple	–	110.6	40.6	33.9	32.6	35.6	35.7	105.1	47.4	129.7	34.4	125.3	24.2	9.6
Average CLE	15.7	53.4	23.4	44.0	55.9	58.5	52.7	38.8	27.4	48.2	44.1	56.8	18.1	5.3

In our all experiments, regularization constant λ is set to 0.01. We resize the target image patch to 32×32 pixels and extract raw feature to represent a target object.

4.2 Quantitative analysis

We perform experiments on ten publicly available standard video sequences. As the ground truth, the center position of a target in a sequence is labeled manually. This ground truth is provided in Wus work.[31] For quantitative analysis, we use average center location errors as evaluation criteria

Table 3 Success rate (SR). Red fonts indicate the best performance while the blue fonts indicate the second best ones. (Ours^{-} represents our tracking method without the forward-backward tracking criterion).

Sequence	PN	L1	VTD	MIL	Frag	CT	WMIL	IVT	MTT	LSAT	LOT	ODOT	Ours^{-}	Ours
DavidIndoor	0.60	0.63	0.53	0.45	0.19	0.50	0.48	0.71	0.53	0.72	0.21	0.14	0.47	0.78
Occlusion1	0.65	0.87	0.77	0.59	0.89	0.71	0.68	0.85	0.79	0.90	0.54	0.87	0.85	0.91
Occlusion2	0.49	0.67	0.59	0.61	0.60	0.59	0.59	0.59	0.72	0.33	0.42	0.68	0.72	0.75
Caviar1	0.70	0.28	0.83	0.25	0.68	0.50	0.42	0.28	0.45	0.85	0.76	0.19	0.27	0.84
Caviar2	0.65	0.81	0.67	0.25	0.26	0.31	0.26	0.59	0.33	0.28	0.80	0.76	0.79	0.83
Caviar3	–	0.42	0.15	0.16	0.13	0.23	0.20	0.13	0.14	0.28	0.23	0.24	0.14	0.82
Car4	0.64	0.84	0.73	0.34	0.22	0.17	0.23	0.92	0.53	0.91	0.18	0.21	0.91	0.92
Car11	0.38	0.43	0.43	0.17	0.08	0.01	0.02	0.74	0.58	0.49	0.37	0.54	0.78	0.81
Deer	0.41	0.04	0.58	0.21	0.07	0.08	0.44	0.22	0.60	0.35	0.55	0.03	0.60	0.62
Jumping	0.69	0.09	0.08	0.51	0.14	0.04	0.02	0.28	0.30	0.09	0.61	0.55	0.53	0.67
Girl	0.57	0.33	0.51	0.52	0.69	0.36	0.41	0.43	0.62	0.08	0.72	0.78	0.48	0.80
Couple	–	0.12	0.38	0.41	0.44	0.42	0.45	0.10	0.30	0.08	0.43	0.11	0.63	0.82
Average SR	0.58	0.46	0.45	0.33	0.37	0.33	0.35	0.49	0.49	0.45	0.49	0.43	0.60	0.80

to compare the performance, and the pixel error in every frame is defined as follows.

$$CLE = \sqrt{(x' - x)^2 + (y' - y)^2} \tag{14}$$

where $(x^{'}, y^{'})$ represents the object position obtained by different tracking methods, and (x, y) is the ground truth. The second evaluated metric is the success rate,[32] and the score in every frame is defined as follows.

$$score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)} \tag{15}$$

where ROI_T is the tracking bounding box and ROI_G is the ground truth bounding box. If the $score$ is larger than 0.5 in one frame, the tracking result is considered as a success. Table 2 reports the center location error, where smaller CLE means more accurate tracking results. In Table 2, each row represents the average center location errors of the eight algorithms testing on a certain

video sequence. The number marked in red indicates the best performance in a certain testing sequence, and the number in blue refers to the second best result. Table 3 reports the success rates, where larger average scores mean more accurate results. From Table 2 and Table 3, we can see that our method achieves the best or second best performance compared with L1, CT, MIL, WMIL, Frag, IVT, MTT, LAST, LOT, VTD, ODOT and PN for most of the sequences. Moreover, we draw the error curve according to equation (14) for each video sequence (Fig.4). In addition, Fig.5, Fig.6, Fig.7, Fig.8, Fig.9, Fig.10 and Fig.11 show the screen captures for some of the video clips. More details of experiments are analyzed and discussed in the following subsections. Overall, our method performs favorably against the other state-of-the-art tracking methods.

4.3 Qualitative analysis

Partial occlusion: The objects suffer heavy or longtime partial occlusion, scale change, deformation and rotation in sequences Caviar1 (Fig.5(a)), Caviar2 (Fig.5(b)), Caviar3 (Fig.5(c)), Girl (Fig.9(a)), Occlusion1 (Fig.7(a)) and Occlusion2 (Fig.7(b)). Fig.5 demonstrates that our tracking method performs well in terms of position and scale when the objects undergo severe occlusion and deformation. In the Caviar1 sequence, our method outperforms all other methods in all given frames, while the MIL, L1, IVT methods completely drift to the background at frames #123, #137, #153, #177, #185, and #195. The CT and WMIL trackers always have some drifts at shown frames. In the Caviar2 sequence, our proposed method can completely track the object when the object suffers partial occlusion at frames #221, while the other methods including the MIL, CT and WMIL completely fail to track the object at frames #223, #317, #331, #456 and #485. In the Caviar3 sequence, the tracked object will be complete occlusion and it has the same color information with the neighbor people. Therefore, it is very difficult to track this object. Our tracker performs better

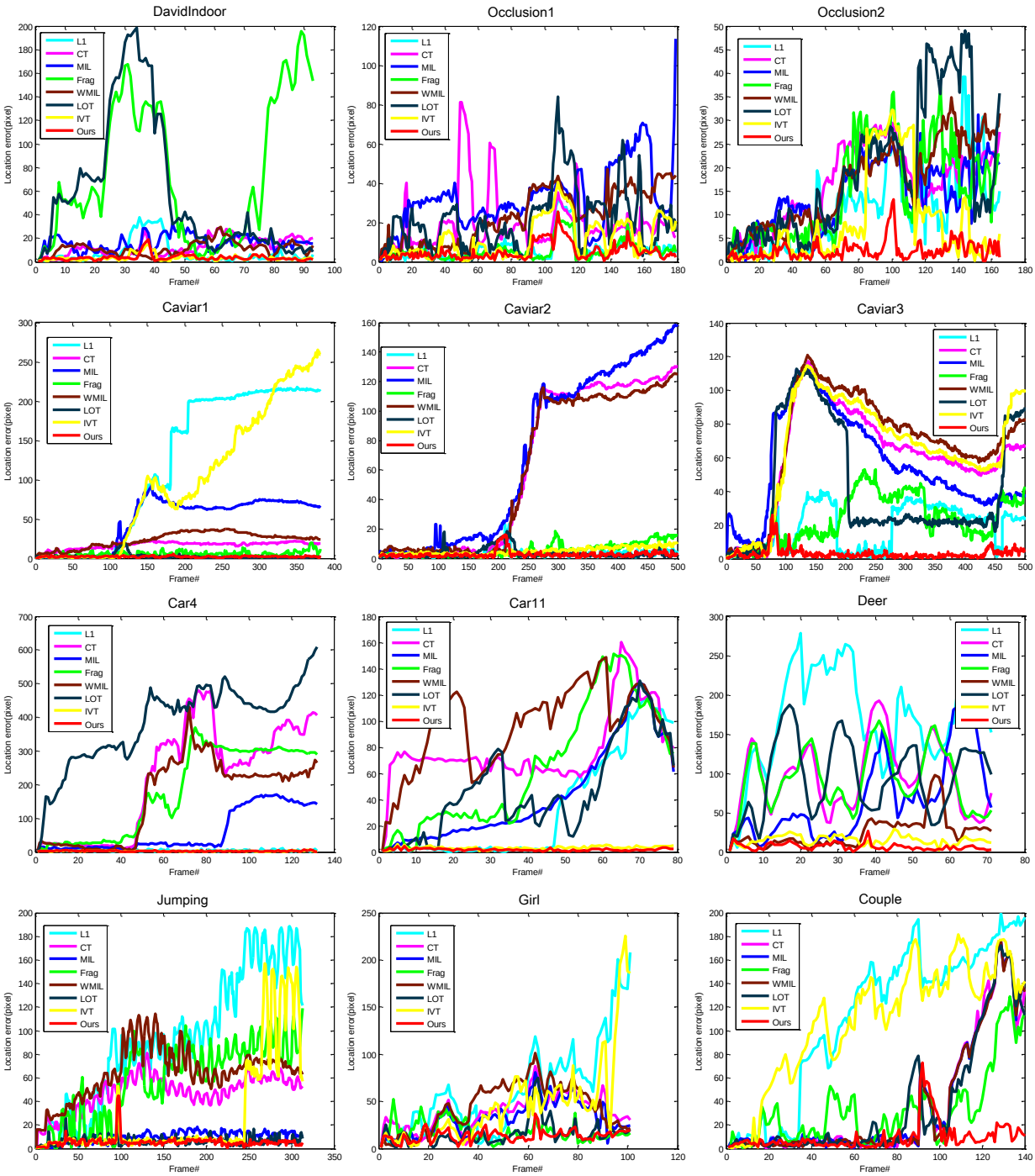


Fig 4 Error plots of all tested sequences for different tracking methods.

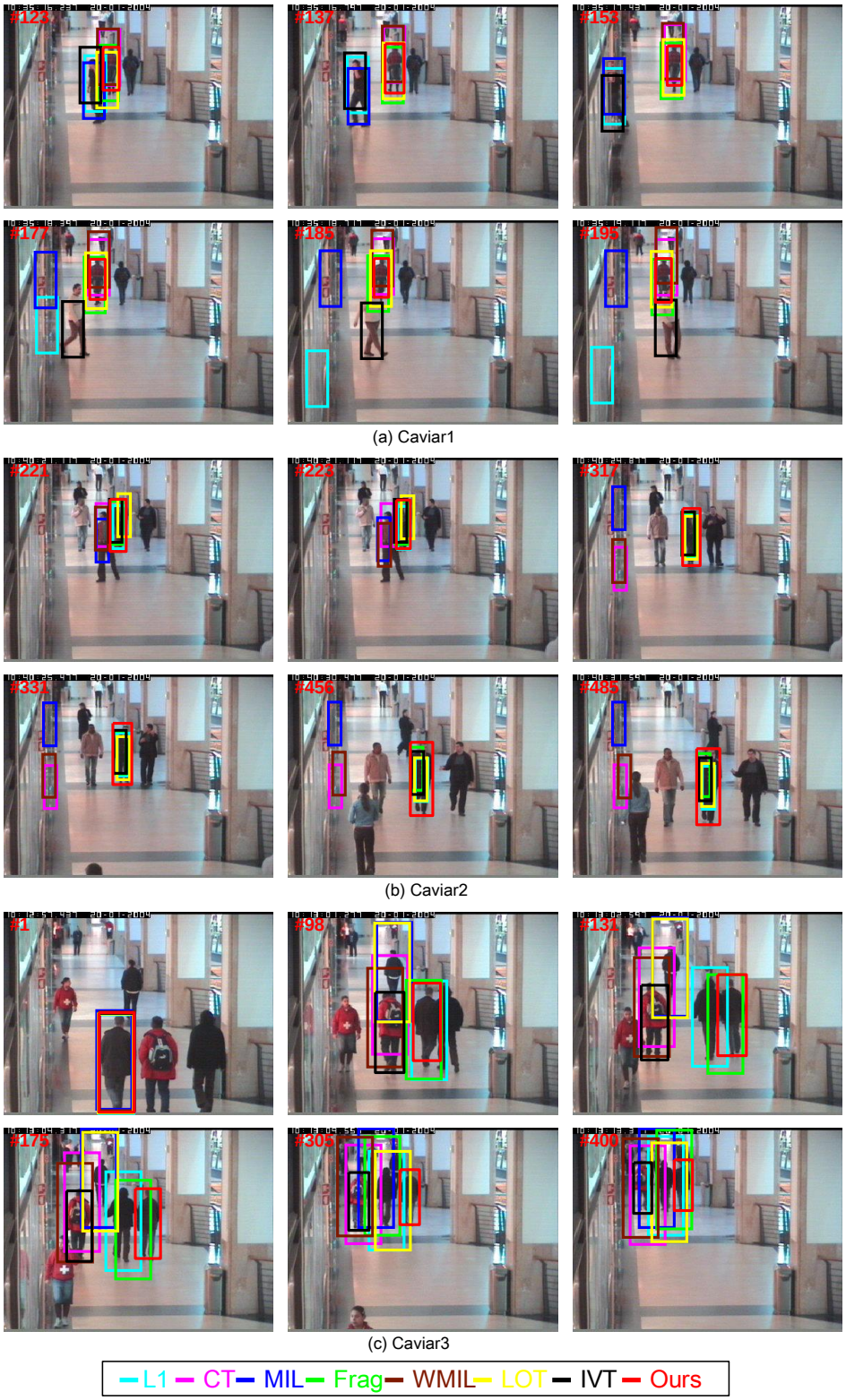


Fig 5 Sampled tracking results for tested sequences of (a) Caviar1, (b) Caviar2 and (c) Caviar3.

than the other all methods whereas the Frag and L1 methods are only able to track the objects at frame #98, but our method can perform more accurately than the two methods at frame #98. What is more, The CT, MIL, LOT, WMIL and IVT methods suffer completely from drift at frames #98, #131, #175, #305 and #400, which verify that the five methods can not adaptively adjust these changes and are not robustness to occlusion, resulting in serious drifts.

In the Occlusion1 and Occlusion2 sequences, we can see our tracking method performs better than the other methods at frames #572, #642, #682, #732, and #822 in Fig 7(a) . The LOT method suffers some drifts at frames #572, #642, #682 and #732. The MIL method also suffers severe drifts at frames #682, #732 and #822. Therefore, these results verify the LOT and MIL methods are not robustness to occlusion. See tracking results in Fig 7(b), our method can track the object very accurately, while the other methods including the LOT, Frag, WMIL, CT, and MIL fail to track the object at frames #581, #626 and #706.

In the Girl sequences, we can see our tracking method performs better than the all other methods at shown frames, especial at frames #433 and #442 when the girl suffers severe occlusion. The Frag and LOT can track the target at frames #433 and #442 but with some drifts, while the WMIL, L1 and CT trackers fail to track the target at all shown frames.

Background clutters: The trackers are easily confused an object is very similar to its background. Fig.6(b) and Fig.10 demonstrate the tracking results in the Deer and Car11 sequences with background clutters. Fig.6(b) shows different trackers track a car in the complex background. Thus, it is very difficult to distinguish the object from its background and to keep tracking the object correctly. Comparatively, our method and the IVT exhibit better discriminative ability and outperform other methods at frames #21, #56, #161, #271, #326 and #391. The MIL and WMIL trackers completely drift to the background at frames #271, #326 and #391, which verifies that the

selected features by the MIL and WMIL trackers are less informative than our method. The Frag tracker has severe drifts at all given frames because its template does not update online, making it unable to handle large background clutter. The CT method has severe drifts at all given frames because it only uses compressive features and the Bayesian classifier is sensitive to background clutter. In the Deer sequence, In the Deer sequence, our method outperforms all other methods in all given frames, while other methods including the CT, Frag, L1, and LOT methods fail to track the Deer at frames #5, #40, #45, #60 and #56 in Fig. We can also see the MIL methods completely fail to track at all given frames.

Abrupt motion and blur: The objects in Deer (Fig.10), Jumping sequences (Fig.8(b)) and Couple (Fig.9(b)) have abrupt motions. It is difficult to predict the location of a tracked object when it undergoes an abrupt motion. As illustrated in Fig.10, when an object undergoes an in-plane rotation, all evaluated algorithms except the proposed tracker do not track the object well. We also see that the WMIL method fails to track at frames #40, #45, #50 and #56. The CT, Frag, LOT, L1 and MIL methods suffer completely from drifts to the background at frames #5, #7, #40, #45, #50, and #56. However, the IVT method can track the object accurately except there some errors at frames #45 and #50. In the Jumping sequence, we can see that our method performs better than other all evaluated algorithms (see all shown frames in Fig.10). The CT, L1, Frag, and WMIL methods suffer completely from drifts in the shown frames. The IVT method performs well at some frames, however, it suffers completely from drifts at frames #247, #287 and #296. See from Fig.8, the MIL and LOT method can track face but there some errors at some frames. In the Couple sequence, our method performs well when the target undergo abrupt motion, while all other methods completely fail to track the target at frames #109, #122, #135 and #140.

Blurry images exist in the Deer and Jumping sequence (see Fig.10 and Fig.8(b)), because a fast

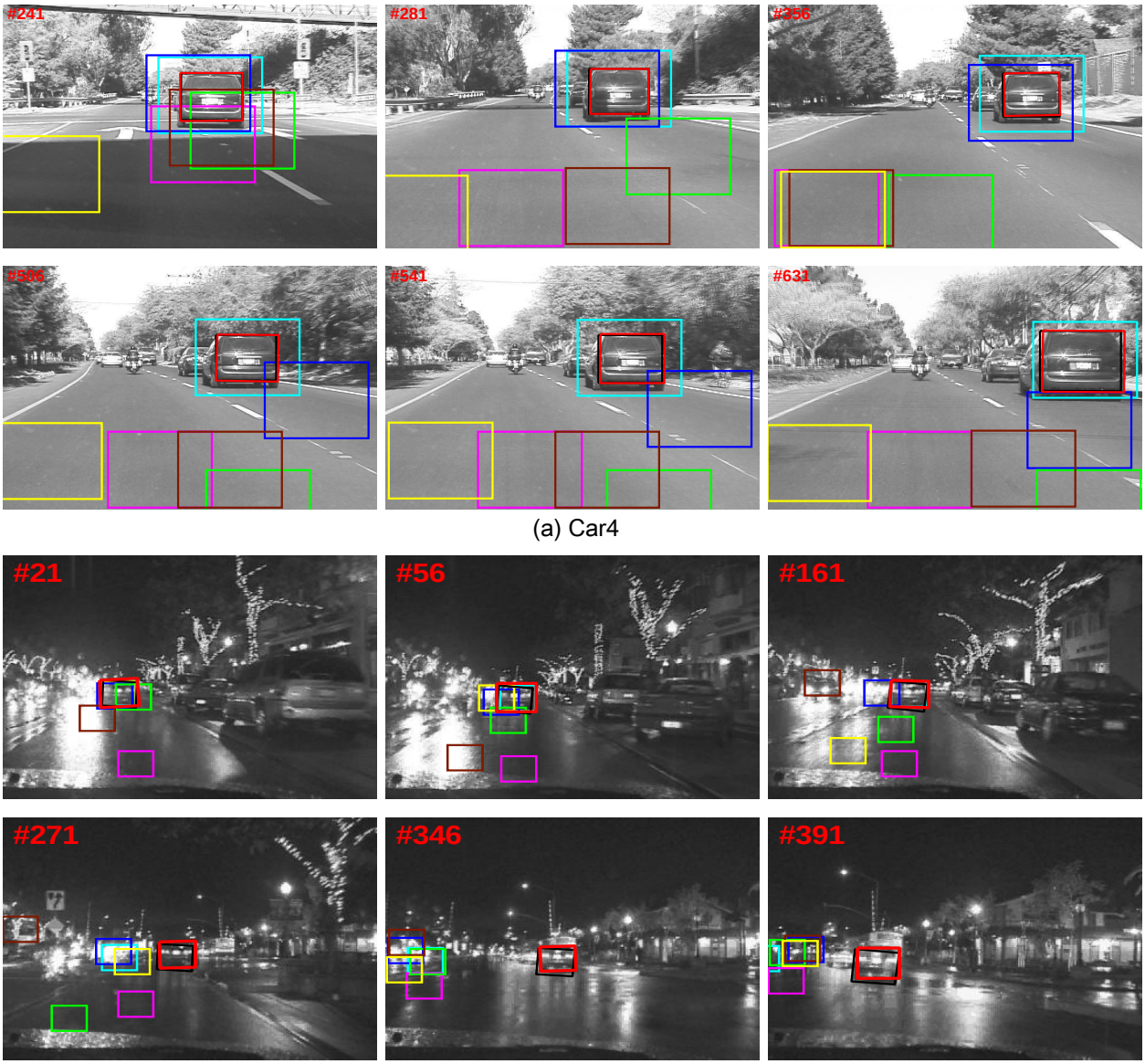


Fig 6 Sampled tracking results for tested sequences of (a) Car4, and (b) Car11.

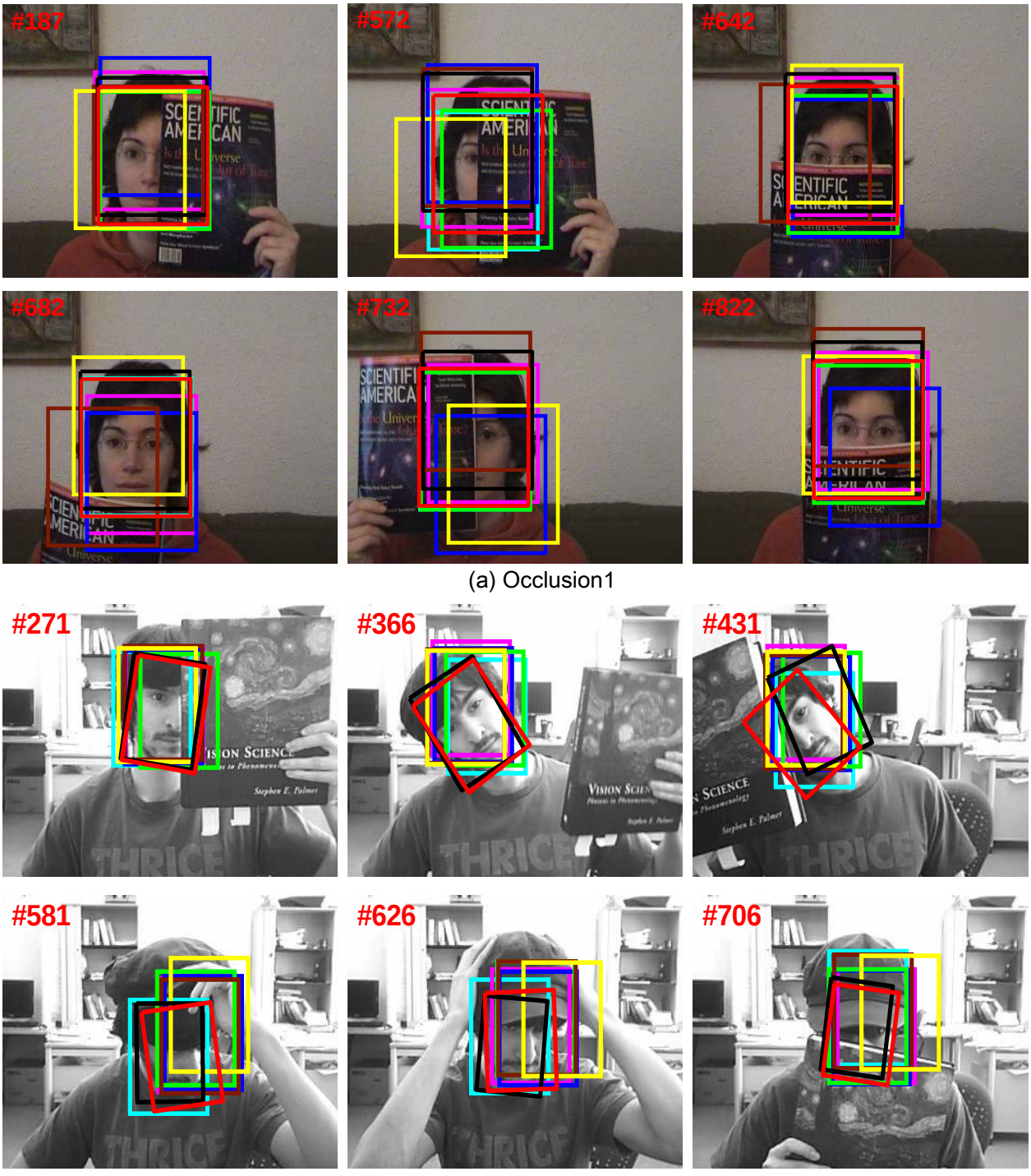
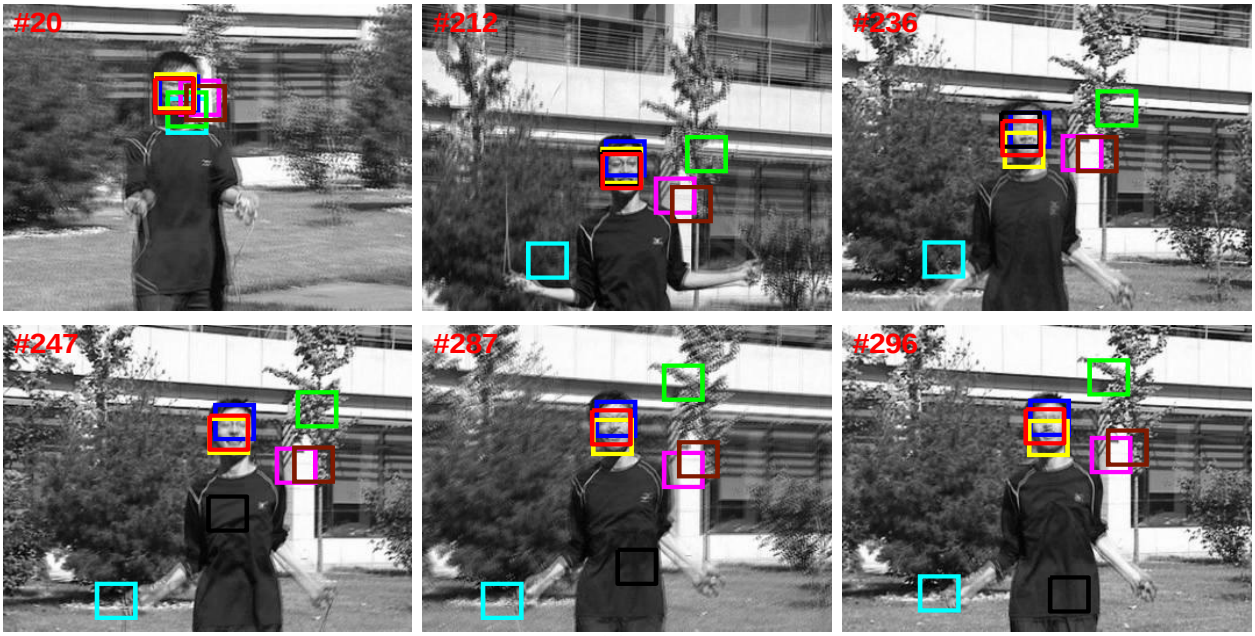


Fig 7 Sampled tracking results for tested sequences of (a) Occlusion1, and (b) Occlusion2.

(a) DavidIndoor



(b) Jumping

L1 CT MIL Frag WMIL LOT IVT Ours

Fig 8 Sampled tracking results for tested sequences of (a) DavidIndoor, and (b) Jumping.

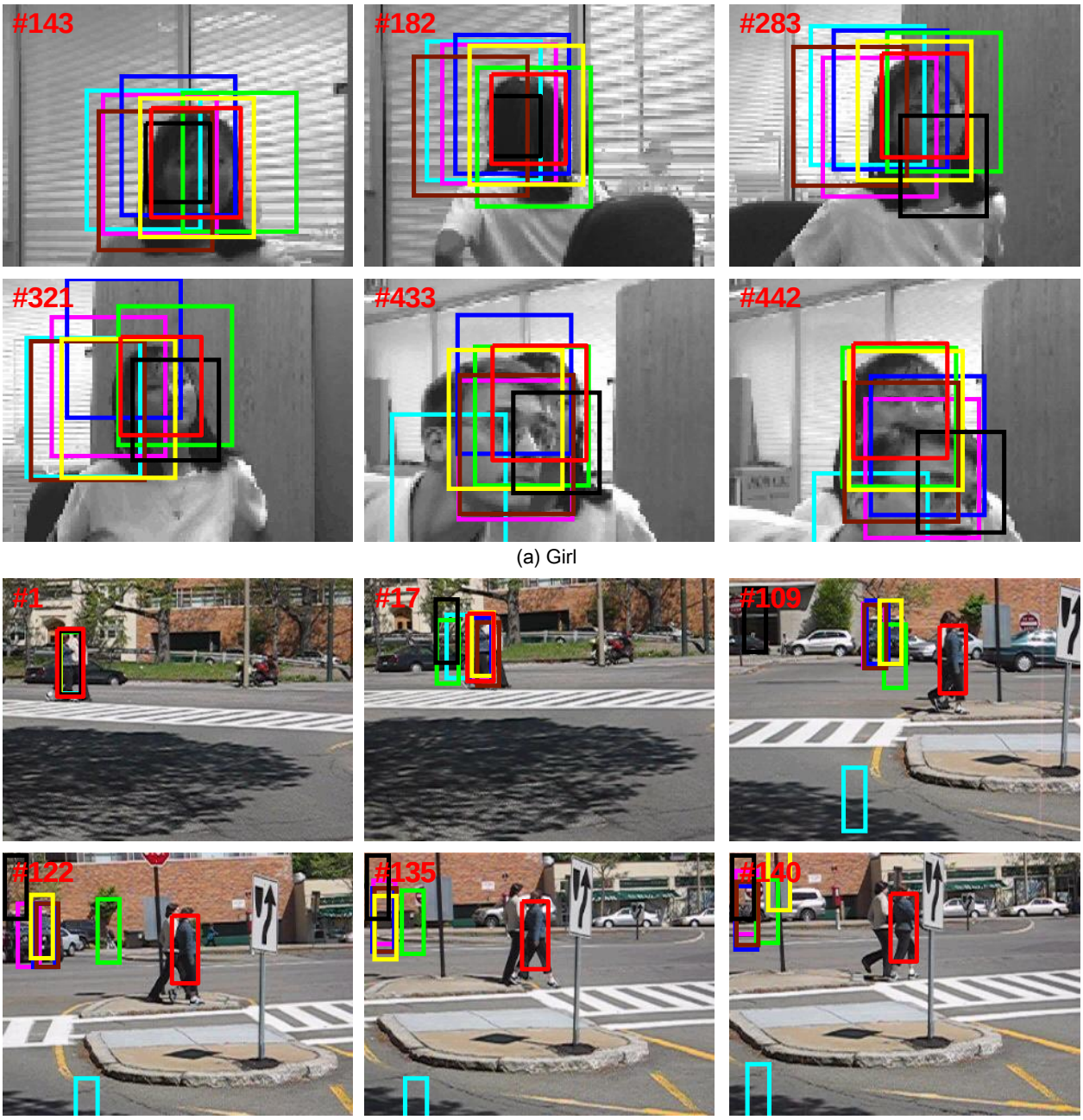


Fig 9 Sampled tracking results for tested sequences of (a) Girl, and (b) Couple.

motion make it difficult to track the target object. As shown in frames #45 and #56 of Fig.10, our proposed method can still track the object well than other methods.

Illumination variation: Fig.6(a), Fig.6(b) and Fig.8(a) show results from four challenging

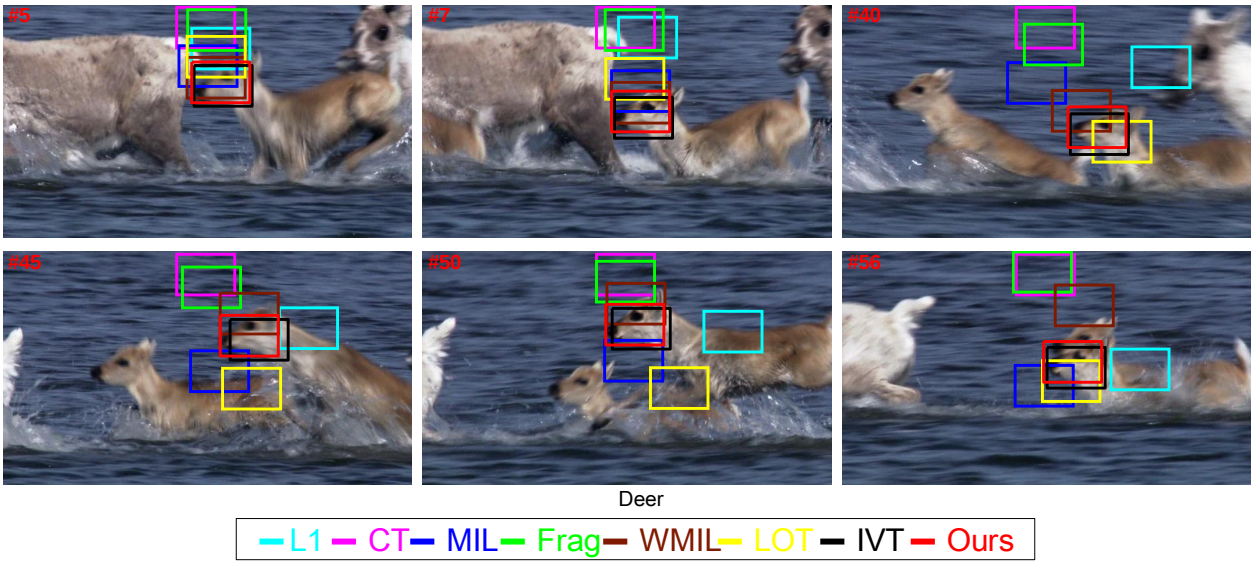


Fig 10 Sampled tracking results for tested sequences of Deer.

sequences with significant change of illumination, scale and pose variation. For the Car 4 sequence, there is a drastic lighting change when the vehicle goes underneath the overpass or the trees. Our method and the IVT can track the car accurately, while the CT, Frag, WMIL and LOT methods suffer completely drifts at the shown frames (in Fig.6(a)). The target object is small with low contrast and drastic illumination change in the Car 11 sequence (Fig.6(b)). Our proposed method and the IVT algorithm perform well in tracking this vehicle whereas the other methods drift away when drastic illumination variation occurs (#200) or when similar objects appear in the scene (#391).

In addition, appearance change caused by scale and pose as well as camera motion pose great challenges. In the DavidIndoor sequence, our method and IVT perform better than the other methods. The Frag tracker suffers completely drifts at the shown frames.

Rotation and shaky factor: The target in the Girl sequence (Fig.9(a)) has big rotation. See from the Girl sequence, the appearance information will change severe when the girl has a rotation. Our method performs well at frames #182 and #283, while the WMIL, L1, CT and MIL trackers

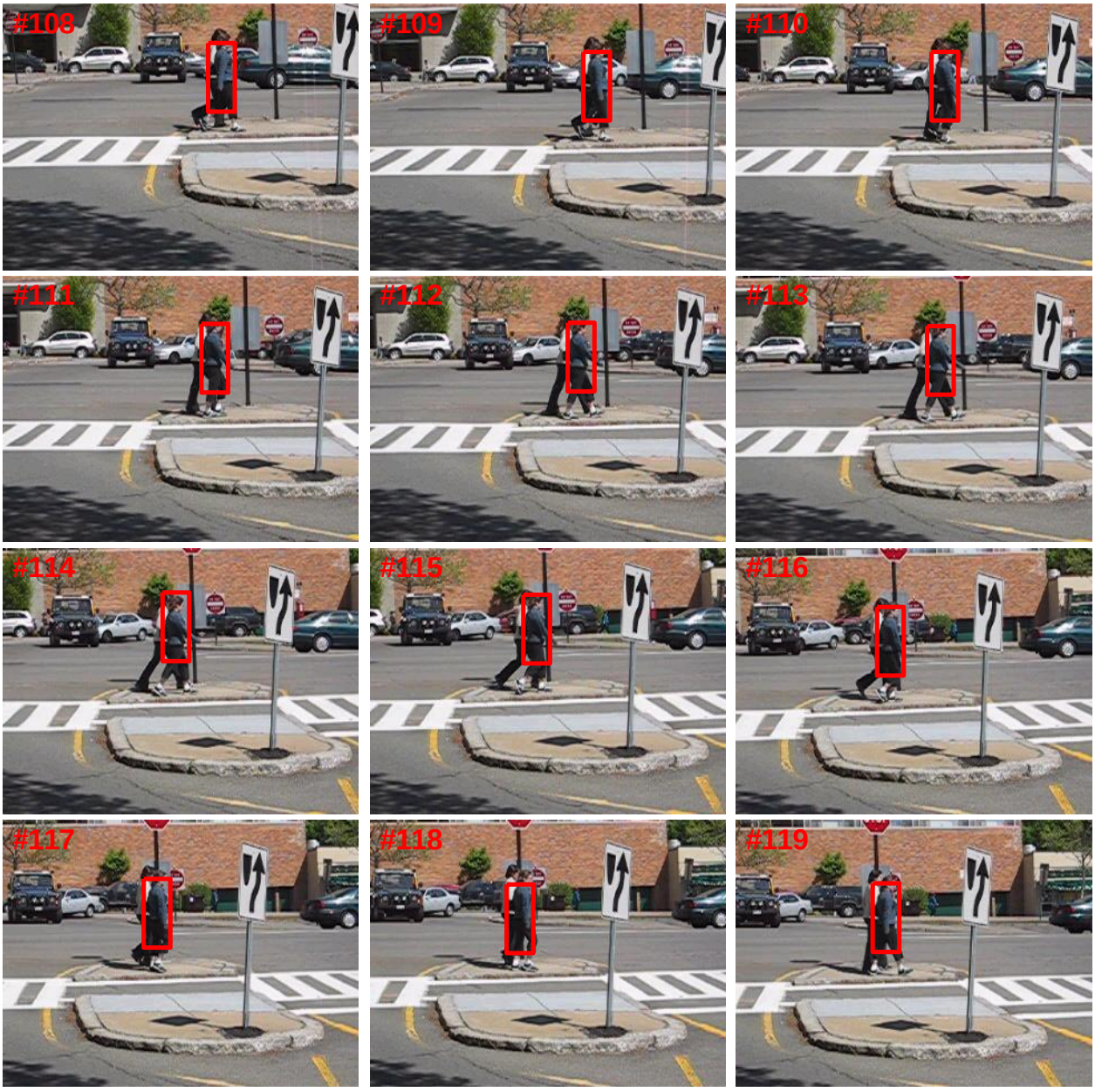


Fig 11 Tracking results of our method on Couple sequence under shaky factor. (from #108 to #109, the sequence shakes up and down; from #109 to #110, the sequence shakes backward; from #113 to #114 and from #115 to #116, the sequence shakes backward sharply)

.

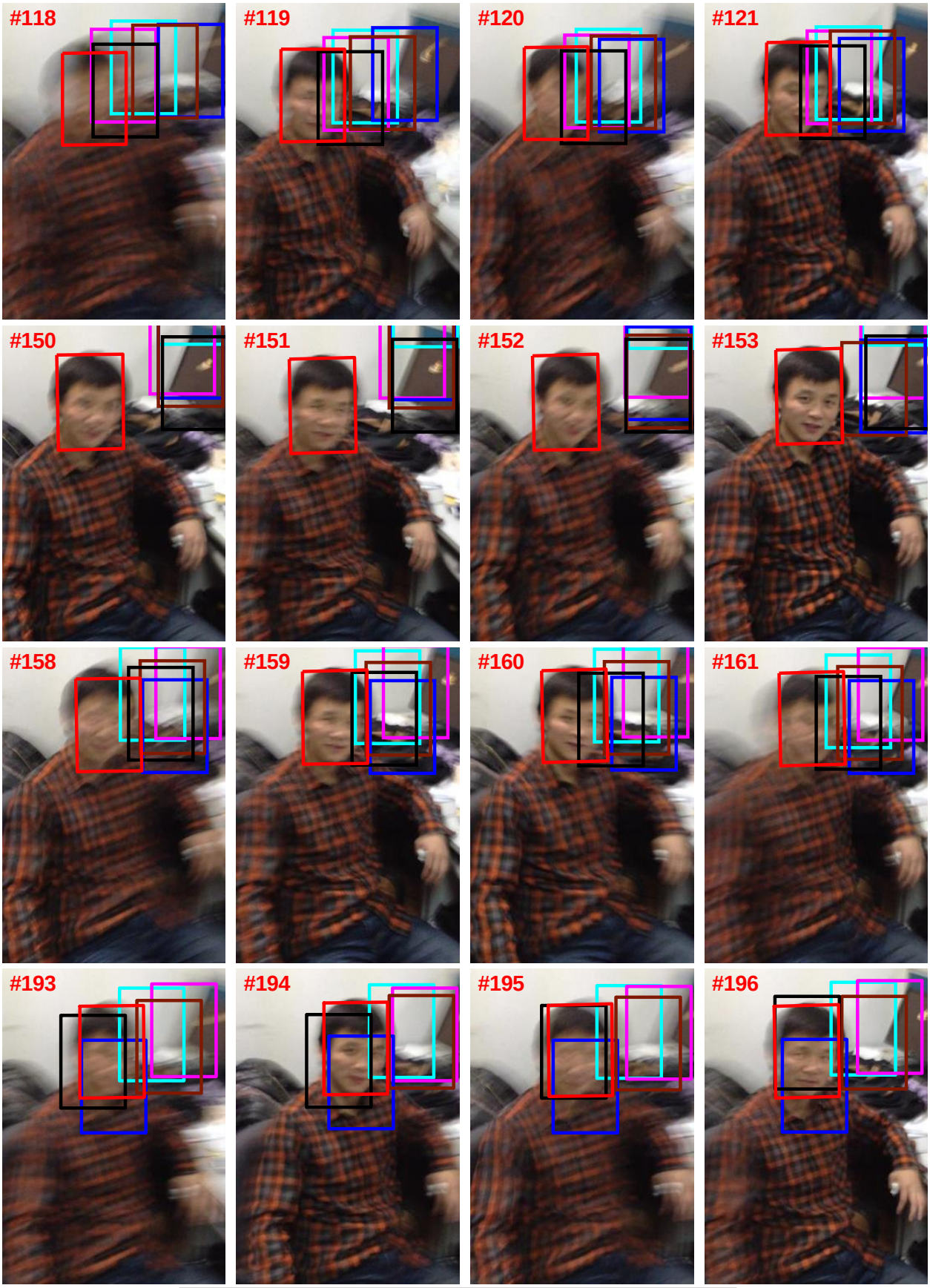


Fig 12 Comparison of tracking results using different trackers on shaky video.

.

suffer from drifts. Shaky factor occurs in the Couple sequence (Fig.9(b)): from #108 to #109, the sequence shakes up and down; from #109 to #110, the sequence shakes backward; from #113 to #114 and from #115 to #116, the sequence shakes backward sharply. As shown in Fig.9(b), our method performs very well under abrupt motion and shaky factor. However, all other methods completely fail to track the target at frames #109, #122, #135 and #140. Fig.11 shows tracking results in details to verify the performance of our method under shaky factor clearly.

In order to further verify the performance of our method under shaky factor, different trackers are tested on the shaky video that is shot in real-world scene (see Fig.12). As shown in Fig.12, the camera quickly moves while the people stays still in the scene, so it brings much motion blur by quick moving. See tracking results from Fig.12, our method performs better than the other methods, especial at frames #118, #120, #150, #158, #161, #193 and #195 suffered from severe motion blur by shaky factor. Therefore, these experimental results show the effectiveness of our method.

4.4 Combined dictionary vs single dictionary

In our method, the discriminative weights are defined using equation (7). The goal is to minimize reconstruction error using positive dictionary while maximizing reconstruction error using negative dictionary. Thus, our method combines positive dictionary and negative dictionary. In order to verify the performance of combined dictionary, we compare our proposed method using combined dictionary, positive dictionary and negative dictionary, respectively. The discriminative weights using positive dictionary only and negative dictionary only are defined as follows:

$$w_i^p = \frac{\xi_i^p}{\sum \xi_i^p} \tag{16}$$

Table 4 Comparison of center location errors against discriminative weights using different dictionary.

Method \ Sequence	Car4	Car11	Deer	Girl	Couple	Jumping	Caviar1	Caviar2	Caviar3	Occlusion1	Occlusion2	DavidIndoor
Positive dictionary	3.2	3.6	9.2	55.3	45.1	4.4	32.5	4.4	62.3	9.0	11.4	43.9
Negative dictionary	3.4	11.1	10.1	37.1	60.5	31.9	81.5	5.7	64.9	90.4	11.7	15.8
Combined dictionary	2.7	1.8	6.8	13.5	9.6	5.5	2.1	2.6	3.0	5.2	6.7	3.5

$$w_i^n = \frac{\frac{1}{\xi_i^n}}{\sum \frac{1}{\xi_i^n}} \tag{17}$$

where $\xi_i^p = \|y_i - D_p\alpha_i^p\|_2^2$ and $\xi_i^n = \|y_i - D_n\alpha_i^n\|_2^2$, D_p and D_n represent positive dictionary and negative dictionary, respectively. The goal is to minimize the reconstruction error using a positive dictionary in equation (16), while the goal is to maximize the reconstruction error using a negative dictionary in equation (17).

In comparison experiments, w_i^p, w_i^n, and w_i represent three different discriminative weights, and they are used to compute the decision map in equation (11). Table 4 reports the center location error using different dictionaries to define the discriminative weights, and Table 5 reports success rates using different dictionaries to define the discriminative weights. From Table 4 and Table 5, we can see that our method combining positive dictionary and negative dictionary achieves the best performance compared with using positive or negative dictionary respectively. These experimental results also verify the distinguish ability of our method using combined dictionary, so our method can distinguish the target effectively from complex background.

Table 5 Comparison of success rates against discriminative weights using different dictionary.

Method \ Sequence	Car4	Car11	Deer	Girl	Couple	Jumping	Caviar1	Caviar2	Caviar3	Occlusion1	Occlusion2	DavidIndoor
Positive dictionary	0.87	0.73	0.60	0.23	0.35	0.70	0.38	0.75	0.15	0.85	0.70	0.31
Negative dictionary	0.84	0.55	0.58	0.55	0.17	0.47	0.27	0.68	0.14	0.36	0.66	0.48
Combined dictionary	0.92	0.81	0.62	0.77	0.79	0.67	0.79	0.83	0.83	0.91	0.75	0.78

4.5 Effect of forward-backward tracking criterion

In some sequences, the target may suffer from occlusion, background clutter, rotation, abrupt motion and other challenging factors. In such cases, many trackers and our method always suffer from drifts or completely fail to track the target. If we directly update the appearance model with the new observation, error is likely to be accumulated and the tracker will drift away from the corrected location. Thus, the forward-backward tracking criterion is used for evaluating the current tracking performance of our method. We update the subspace appearance model using new observations if the forward-backward tracking error is very small.

In order to verify the performance of forward-backward tracking criterion, our method is implemented on some challenging sequences with forward-backward tracking criterion and without. These comparison results are shown in Table 2 and Table 3 (Ours$^-$ represents our tracking method without the forward-backward tracking criterion). Fig.13 shows comparison results of our tracking method using the forward-backward tracking criterion and without the criterion. As shown in Table 2, Table 3 and Fig.13, our method with the forward-backward tracking criterion performs well than without the criterion. Thus, this criterion can effectively evaluate the tracking performance to decide whether or not to update the appearance model.

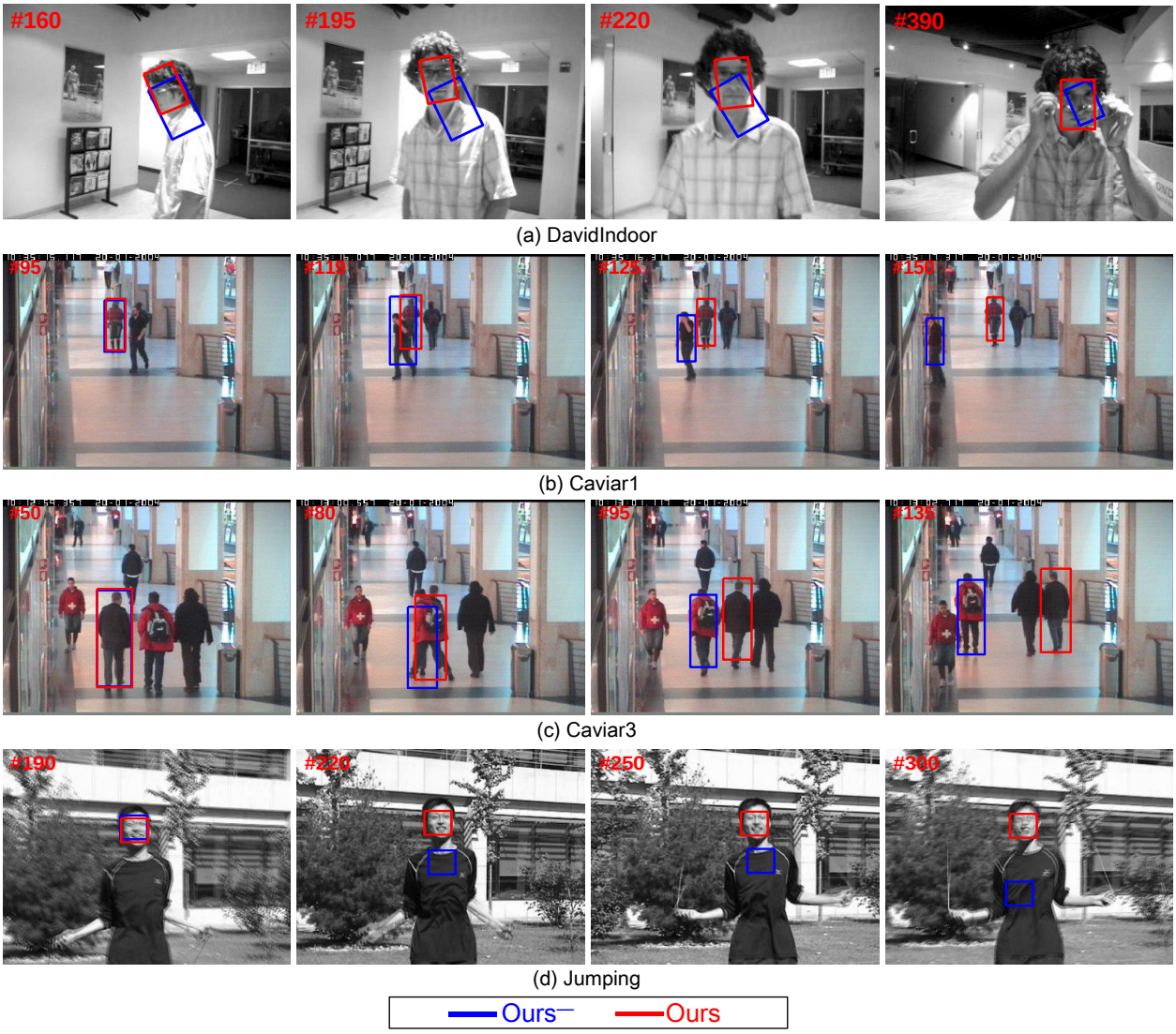


Fig 13 Comparison of our tracking method using the forward-backward tracking criterion and without.

Table 6 Comparison of average FPS.

Algorithm	CT	MIL	IVT	WMIL	LOT	ODOT	Ours
Average FPS	34.3	6.7	20.1	24.4	0.2	0.3	3.7

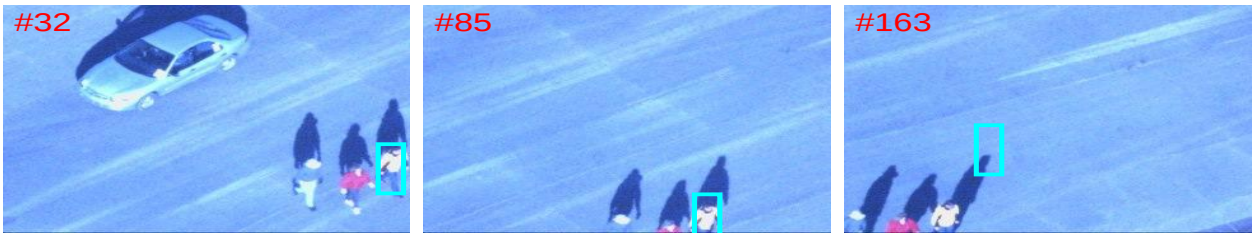
4.6 Complexity analysis

In the IVT method, the computation involves matrix-vector multiplication and the computation complexity is $O(dk)$. The computation complexity of the CT tracker using random projection to extract features is $O(cn)$, where c is the number of nonzero entries in each row of projection matrix. The computation complexity of LASSO algorithm to compute the sparse coefficients for sparse representation is $O(d^2 + dk)$. The ODOT method needs to implement two-stage object tracking using sparse representation, so it is very slow. The computational load of our method is mainly to compute sparse coefficients and subspace appearance model construction, and the complexity is $O(d^2 + dk)$.

In order to compare the detailed computational time of our tracker with other tracking methods, we test different trackers using MATLAB on an i3 3.20 GHz machine with 4 GB RAM. Then, some selected trackers are implemented on different video sequences, the whole running time is stored on each sequence, and then we can obtain the frames per second (FPS) at the all tested sequence. Finally, we report the average FPS from the all test sequences in Table 6.

4.7 Discussion

As shown in our experiments, our method can address these factors including abrupt motion, cluttered background, occlusion, and Illumination variation more effectively. This can be attributed to some reasons listed as follows. (1) We define the discriminative weights through estimating the sparse construction error using negative and positive samples, which help our method to distinguish



(a) Pedestrian2

(b) Biker

Fig 14 Two failed tracking cases:(a) out of plane rotation and abrupt motion; (b) object of interest leaves completely out of screen and reappears.

target from background clutter accurately. (2) The decision map combining discriminative weights and subspace reconstruction error can use the advantages of sparse representation and subspace learning model, which help to handle the appearance change and background clutter effectively. (3) The new valuation criterion based on the forward and backward tracking method can handle tracking outliers and reduce the cumulative error. Therefore, our tracker can obtain favorable performance.

However, our proposed method may fail when an object of interest leaves completely out of screen and reappears or an out-of-plane rotation and an abrupt motion occur in the current sequences (see Fig.14). Fig.14(a) shows the tracked object completely out of the screen and reappears after some frames. Our tracker can not track the object in a long time when an object of interest leaves completely out of screen, so there are big errors to update the subspace appearance model. Fig.14(b) shows an out-of-plane rotation and an abrupt motion after #69. Our method drifts away the ground truth because the appearance model can not match well between the object model and the candidates, and it cannot distinguish the object from the changed background when abrupt

motion.

Overall, our method performs favorably against the other state-of-the-art tracking methods in the challenge sequences.

5 Conclusion

In this paper, we have proposed a novel tracking algorithm based on a weighted subspace reconstruction error. Firstly, the discriminative weights are defined through minimizing the reconstruction error using a positive dictionary while maximizing the reconstruction error using a negative dictionary respectively. The discriminative weights can distinguish a target from its background clutter accurately due to the use of positive and negative samples to encode sparse coefficients. Combining discriminative weights and subspace reconstruction error can make use of their advantages including sparse representation and subspace appearance model, which help to handle appearance variation and sever occlusion effectively. Furthermore, the new valuation method based on forward-backward tracking criterion can handle tracking outliers and reduce the cumulative error. Experiments on some challenging video sequences have demonstrated the superiority of our proposed method to twelve state-of-the-art ones in accuracy and robustness.

Acknowledgments

This research is partly supported by NSFC, China (No: 61273258and Shanghai SAST funding.

References

1 A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, **1**, 798–805, IEEE (2006).

2 J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1269–1276, IEEE (2010).

3 X. Mei and H. Ling, "Robust visual tracking using 1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*, 1436–1443, IEEE (2009).

4 D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision* **77**(1-3), 125–141 (2008).

5 M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision* **26**(1), 63–84 (1998).

6 D. Chen and L. Yang, "Robust object tracking via online dynamic spatial bias appearance models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **29**(12), 2157–2169 (2007).

7 T. Zhou, X. He, K. Xie, K. Fu, J. Zhang, and J. Yang, "Visual tracking via graph-based efficient manifold ranking with low-dimensional compressive features," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, 1–6, IEEE (2014).

8 B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 983–990, IEEE (2009).

9 K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision–ECCV 2012*, 864–877, Springer (2012).

10 Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(7), 1409–1422 (2012).

11 K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition* (2012).

12 K. Fu, C. Gong, Y. Qiao, J. Yang, and I. Y.-H. Gu, "One-class support vector machine-assisted robust tracking," *Journal of Electronic Imaging* **22**(2), 023002–023002 (2013).

13 A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **25**(10), 1296–1311 (2003).

14 T. Bai and Y. F. Li, "Robust visual tracking with structured sparse representation appearance model," *Pattern Recognition* **45**(6), 2390–2404 (2012).

15 R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(10), 1631–1643 (2005).

16 H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting.," in *BMVC*, **1**(5), 6 (2006).

17 H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Computer Vision–ECCV 2008*, 234–247, Springer (2008).

18 C. Zhang, R. Liu, T. Qiu, and Z. Su, "Robust visual tracking via incremental low-rank features learning," *Neurocomputing* (2013).

19 Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, 425–432, IEEE (2012).

20 Y. Hou, W. Li, A. Rong, H. Lou, and S. Quan, "Robust visual 2-regularized least squares tracker with bayes classifier and coding error," *Journal of Electronic Imaging* **22**(4), 043036–043036 (2013).

21 Z. Hong, X. Mei, D. Prokhorov, and D. Tao, "Tracking via robust multi-task multi-view joint sparse representation," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, 649–656, IEEE (2013).

22 Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1697–1704, IEEE (2011).

23 B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Computer Vision–ECCV 2010*, 624–637, Springer (2010).

24 T. Zhou, J. Zhang, K. Xie, J. Yang, and X. He, "Visual tracking based on weighted subspace reconstruction error," in *IEEE International Conference on Image Processing 2014 (ICIP 2014)*, (Paris, France) (2014).

25 S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, 41–48, IEEE (1999).

26 Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, 2756–2759, IEEE (2010).

27 T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse

learning," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2042–2049, IEEE (2012).

28 B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1313–1320, IEEE (2011).

29 Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 49–56, IEEE (2010).

30 S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 1940–1947, IEEE (2012).

31 Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2411–2418, IEEE (2013).

32 M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision* **88**(2), 303–338 (2010).

List of Figures

1 The flow of our proposed tracking algorithm.

2 Illustration of details to define the discriminative weights. It is similar to the linear discriminant criterion: the goal is to minimize the reconstruction error using a positive dictionary while maximizing the reconstruction error using a negative dictionary. The discriminative weights are obtained by the positive and negative dictionary reconstructions respectively and can ensure the distinguish ability.

3 Illustration of evaluation criterion via forward and backward tracking method.

4 Error plots of all tested sequences for different tracking methods.

5 Sampled tracking results for tested sequences of (a) Caviar1, (b) Caviar2 and (c) Caviar3.

6 Sampled tracking results for tested sequences of (a) Car4, and (b) Car11.

7 Sampled tracking results for tested sequences of (a) Occlusion1, and (b) Occlusion2.

8 Sampled tracking results for tested sequences of (a) DavidIndoor, and (b) Jumping.

9 Sampled tracking results for tested sequences of (a) Girl, and (b) Couple.

10 Sampled tracking results for tested sequences of Deer.

11 Tracking results of our method on Couple sequence under shaky factor. (from #108 to #109, the sequence shakes up and down; from #109 to #110, the sequence shakes backward; from #113 to #114 and from #115 to #116, the sequence shakes backward sharply)

12 Comparison of tracking results using different trackers on shaky video.

13 Comparison of our tracking method using the forward-backward tracking criterion and without.

14 Two failed tracking cases:(a) out of plane rotation and abrupt motion; (b) object of interest leaves completely out of screen and reappears.

List of Tables

1 Evaluated video sequences.

2 Center location error (CLE). Red fonts indicate the best performance while the blue fonts indicate the second best ones. (Ours$^-$ represents our tracking method without the forward-backward tracking criterion).

3 Success rate (SR). Red fonts indicate the best performance while the blue fonts indicate the second best ones. (Ours$^-$ represents our tracking method without the forward-backward tracking criterion).

4 Comparison of center location errors against discriminative weights using different dictionary.

5 Comparison of success rates against discriminative weights using different dictionary.

6 Comparison of average FPS.