Wang *et al. EURASIP Journal on Advances in Signal Processing* (2019) 2019:50
https://doi.org/10.1186/s13634-019-0638-0

EURASIP Journal on Advances in Signal Processing

REVIEW Open Access

Online non-negative discriminative dictionary learning for tracking

Weisong Wang[1], Fei Yang[1*] and Hongzhi Zhang[2]

Abstract

In this paper, online non-negative discriminative dictionary learning for tracking is proposed, which combines the advantages of the global dictionary learning model and the class-specific dictionary learning model. The previous algorithm based on general dictionary learning does not take into account the inter-class relations between classes and make full use of tag information. In order to improve the classification ability of dictionaries, the class correlation was proposed to guide the learning of discriminant dictionaries, which makes full use of the correlation and difference between the atomic classes of dictionaries and introduces the tag information of the categories to improve the discriminant ability of dictionaries. For this purpose, the Huber loss function and the Fisher weight coefficient is used in the discriminative term to improve computational efficiency. In addition, non-negative constraints is added on dictionaries to enhance the performance. The OTB50 and OTB100 datasets are used to evaluate our tracker and compare with related algorithm. The experimental results show that our method performs much better than the tracking method compared in this paper.

Keywords: Object tracking, Discriminative learning, Dictionary learning, Sparse coding

1 Introduction

With the continuous development of computer hardware and artificial intelligence algorithms in the resent years, using machines to assist or partially replace human is the current trend in the field of information technology. As one of the basic research areas of computer vision, target tracking method can provide computers with important target motion-related information, which helps computers to analyze and understand the behavior of targets to make decisions and actions. Although there are many requirements for target tracking, the difficult factors of target tracking are doubled due to complicated scene and diversified requirements. During tracking, the appearance of the target may change drastically due to the rotate, occlusion, etc. There are also dramatic changes in lighting, rapid movement of the target, camera-shake, and other situations, which make the target cannot be well displayed in the image sequences. In practice, all need to be trade-off between software and hardware environment, speed requirements, accuracy and robustness of the algorithm that further increases the difficulty of the algorithm research. Especially , those problems are huge challenge for the UAV target tracking [1, 2] .

In recent years, experts, scholars, and engineers have invested a lot of research, proposed a variety of single target tracking algorithms, and builded various databases in order to solve these problems. Smeulders et al. [3] summarized the 19 most representative algorithms of nearly 10 years. Those algorithms were divided into five classes: matching, matching with extended appearance, matching with constraints, discriminative classification, discriminative classification with constraints. In the recent research literatures [4–7], the tracking methods are classified generative model and discriminative model.

Generative tracking methods describe the target appearances using generative models and search for the target regions that fit the models best. In general, the target model is represented by a subspace or a basis vector consisting of a series of templates. In order to better learn object appearance, Ross et al. [8] proposed incremental visual tracking (IVT) which used principal component analysis (PCA) to learn a low-dimensional subspace representation, and online updated the target changes. In

*Correspondence: feiyang@sdu.edu.cn
[1]Shandong University at Weihai, School of Mechanical, Electrical and Information Engineering, Wenhua Xilu, 264209 Weihai, China
Full list of author information is available at the end of the article

the same year, Han et al. [9] applied the mixed probability density estimation model to the target tracking algorithm and worked well. In 2010, Kwon et al. [10] proposed visual tracking decomposition (VTD) model which provided an efficient strategy for dividing tracking problem into basic observation model and motion model. Visual tracking decomposition method integrates multiple basic trackers into one robust compound tracker while interactively improves the performance of all basic trackers. Among the generative tracking methods, the most representative algorithm is the tracking algorithm based on the sparse representation. Mei et al. [11, 12] proposed L1T algorithm of sparse representation with ℓ_1 norm to reduce the effect of object internal factor (such as rotation, scale transform) and object external factor (such as illumination) change. Extracting features based on the appearance model from the data-independent multi-scale image space was used by Zhong et al. [13] to improve the efficiency of the algorithm. That same year, Zhang et al. found that the potential relationship between the sampling particles can improve the performance of the tracking algorithm, and then they proposed the compressive tracking (CT) [14]. Luka and Matej [15] proposed a coupled-layer visual model optimization method to solve rapid and significant appearance changes. Zhou et al. [16] proposed sparse heterogeneous feature representation (SHFR) for multi-class heterogeneous domain adaptation (HDA) to learn a sparse feature transformation between domains with multiple classes.

Different from the generative tracking methods, discriminative tracking methods treat the target tracking process as a binary classification problem. The methods use a classifier to separate the target from the background. Babenko et al. [17] used online multi-instance learning to capture positive and negative samples with uncertainty as a classification algorithm for target tracking. Kalal first [18] used unlabeled structured data and a semi-supervised learning algorithm to design online tracking method. Tracking learning detection (TLD) [19] algorithm was proposed by adding redetection after the tracking failed. Subsequently, Hare proposed Struck [20] algorithm which used online tructured output based on support vector machine. The MIL tracker [21] integrated the sample importance into an efficient online learning to improve the preformance of classifier. Among discriminative tracking algorithms, the tracking algorithm based on correlation filter (CF) stood out and developed rapidly with its high speed and high efficiency. Bolme et al. [22, 23] proposed MOSSE (minimum output the sum of squared error) algorithm based on correlation filter, which transformed the image from spatial domain to frequency domain, greatly reduced the memory requirements and computational burden. João et al. [24] firstly introduced cyclic matrix into the visual tracking method based on correlation filter, and then the tracking method by linear space was extended to nonlinear space in [25]. Yao et al. [26] proposed RTINet approach for joint off-line training of deep representation and model adaptation in CF trackers.

However, in complex environment, the discriminative tracking algorithm can perform better. This is due to the use of negative samples in the discriminative model, which can avoid drifting in the tracking process. Generally speaking, combining the two models can achieve better results than a single model. Wang et al. [4] proposed the method of online non-negative dictionary learning (ONNDL), which was a good combination of the generative model and the discriminative model. Yang et al. [27] combined dictionary learning with positive and negative label information and proposed an online discriminative dictionary learning tracking method.

In this paper, online non-negative discriminative dictionary learning for tracking (ONDDLT) algorithm is proposed, which combines the advantages of the global dictionary learning model and the class-specific dictionary learning model. The contributions of this paper are summarized as follows:

- To solve the residual growth problem of objective function and improve the robustness of matrix of singular value, the ℓ_1 norm is replaced by Huber loss function.
- Fisher weight coefficient is used to replace the support vector algorithm of adaptive weight coefficient in the discriminative term in order to make the objective function easier to solve.
- Non-negative constraints on dictionaries are added to enhance the interpretability and system performance.
- Experimental results on the tracking benchmark shows that our tracker achieves the first tracking performance compared with other methods based on sparse coding in this paper.

2 Related work

2.1 The appearance representation in tracking

It's difficult to solve how to use the appearance of the target object and its features to represent the target in the visual tracking. In the current research of tracking algorithms, different ideas and methods are proposed to solve the problem of object appearance representation. In [4, 11, 12, 14, 15, 27, 28], the target object image was used as the dictionary atom after feature extraction, and the new target image was used to update the dictionary in the tracking process, so as to reconstruct the apparent model of the target in different periods of time. In [5, 29, 30], in order to cope with the gesture change and occlusion of the target object well, the target was divided into multiple parts. Feature extraction is also one of the important ways of object representation. In [30], color histogram

statistics in color space were used as the representation characteristics of the target. In [31], a variety of usual feature extraction methods were used to combine and form new features by utilizing complementary information between features. In the research [6, 7, 32–35], deep leaning (DL) was used as the extraction method of tracking algorithm and obtained great success. Wang et al. [32] proposed the point-to-set distance metric learning which was conducted on convolutional neural network features of the training data extracted from the starting frames. Lei Qu et al. [36] integrated fast histogram of oriented gradient (FHOG) and discriminative color descriptors (DD) to further boost the tracking performance.

2.2 Discriminative dictionary learning

The goal of discriminative dictionary learning is to enhance the discriminative ability of the coefficient vector while learning the dictionary. There are two learning strategies: the global dictionary learning model and the independent dictionary learning model. The global dictionary learning model is to learn a dictionary whose atom corresponds to all categories of the training set. Mairal et al. [37] explored the structured information of the dictionary through the classifier trained by the coefficient vector, thereby improving the recognition and classification ability of the discrimination dictionary. Pham et al. [38] proposed joint optimization K-SVD face recognition discriminant dictionary learning. In [39], linear SVM (support vector machine) was used to simultaneously optimize the dictionary and classifier that made the dictionary and coefficient vector more adaptive and flexible. These global dictionary learning could use a small dictionary to represent the training data but they ignored the relationship between the category label and the dictionary atom. The independent dictionary learning model means that each class corresponds to a single dictionary and each dictionary atom corresponds to only one class. Structured dictionary learning model proposed by Ramirez et al. [40] could improve the discriminative ability of subdictionaries between different categories. In [41], author proposed an unified joint discriminative feature learning framework in which uncontaminated and corrupted features, classier parameters of multiple visual cues. This paper [42] proposed to jointly learn heterogeneous features and classifiers for multi-modality tracking under discriminabilty-consistency constraint. In [43], they proposed to extract informative feature templates and exploit the modality consistency in discriminability and representation ability for modality fusion-based appearance modeling. Yang et al. [44] explored the Fisher discriminant criterion to learn the discriminant dictionary.

2.3 Tracking algorithm based on dictionary learning

The online dictionary learning tracking method is the target tracking method based on sparse coding technology. Different from general sparse coding, the training samples of the target template dictionaries are increasing, and the dictionaries are required to maintain a high update speed. Accordingly, the online dictionary learning algorithm can reduce the update time of the general dictionary, so as to meet the online target tracking method's demand for the update speed as much as possible. In L1T [11], the basis vector which was made up of the target template and the minor template was used to describe the target. The linear combination of the sparse basis vector was used to reconstruct the candidate region particles. The target template corresponded to the appearance of the target. The minor template was mainly used to deal with noise and occlusion. Zhang et al. [38] proposed a novel tracking model which used a semi-supervised appearance dictionary learning method. In general, a small number of minor template could significantly reduce the reconstruction error. In the online non-negative dictionary learning target tracking method (ONNDL) [4], the Huber loss function was used to instead of the minor template, thereby reducing the calculation consumption. Mathematically, it is correct to have negative values in the decomposition results from a computational point of view, but negative elements are often meaningless in practical problems. This is why non-negative constraint able to enhance the tracking performance. Both [45–48] were related with sparse coding. The method of sparse coding combined with non-negative constraint could improve the robustness and accuracy of the model. Sparse dictionary learning as same as sparse coding could combined with non-negative constraint. In addition, the dictionary learning method of mapping gradient descent model was adopted to solve the problem of online dictionary learning.

3 Proposed method

3.1 The objective function of algorithm

In general discriminant dictionary learning, training samples and their labels are all known in advance, and dictionaries can be fully learned through corresponding training. In the process of target tracking, the results of each tracking provide new training samples and labels, and the dictionary is constantly updated in the process. Accordingly, we adopt the same method as Wang [4], mapping gradient descent method, which can make the dictionary faster and better. The Huber loss function is slower than the ℓ_2 norm when the residuals increase, which is conducive to the robustness of singular values. Therefore, in the target function, the Huber loss function is used to replace the norm as the reconstruction term of dictionary learning. At the same time, the $\ell_{1,\infty}$ norm is used as regularization term inspired by the class correlation. It can fully use the inter-class relations and tag information within the class. In conclusion, the objective function is shown in Eq. (1).

$$\min_{\boldsymbol{D},\boldsymbol{A}} f(\boldsymbol{D},\boldsymbol{A}) = \sum_i \sum_j \ell_\delta(x_{ij} - \boldsymbol{d}_{\cdot i}\boldsymbol{a}_{\cdot j}) + \gamma \sum_c \|\boldsymbol{A}_c\|_{1,\infty} + \frac{\beta}{2}\mathcal{L}(\boldsymbol{A},\boldsymbol{W}) s.t. \boldsymbol{D} \geqslant 0, \boldsymbol{A} \geqslant 0, \boldsymbol{d}_{\cdot k}^T \boldsymbol{d}_{\cdot k} \leqslant 1, \forall k. \tag{1}$$

where $\boldsymbol{D}$ is the matrix of dictionary template, $\boldsymbol{A}$ is the matrix of expression coefficient, x_{ij} is the element of row and column of training sample, $\boldsymbol{d}_{\cdot}$ is the vector of dictionary template and $\boldsymbol{a}_{\cdot}$ is the vector of expression coefficient. $\mathcal{L}(\boldsymbol{A},\boldsymbol{W})$ is the discriminative term. $\boldsymbol{W}$ is the weight the weight coefficient matrix of the discriminative terms. $\boldsymbol{A} \geqslant 0$ is to make sure the coefficients are not negative. γ and β are parameters that can be set manually to adjust the effects of the regularization term and the discriminative term. $\ell_\delta(\cdot)$ represents the Huber loss function. The specific form is shown in Eq. (2).

$$\ell_\delta(\mathrm{r}) = \begin{cases} \frac{1}{2}r^2 & |r| < \delta \\ \delta|r| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{2}$$

where δ is the parameter of Huber loss, and it controls the velocity of gradient descent. In the previous paper, the support vector of the weight coefficient was used to represent the discriminative term. The solution of Huber loss function is too complex. In the method of Cai [39], the Fisher discriminative can be simplified into Eq. (3). According

to the Fisher discrimination criterion, a structured dictionary, whose dictionary atoms have correspondence to the class labels, is learned so that the reconstruction error after sparse coding can be used for pattern classification. Meanwhile, the Fisher discrimination criterion is imposed on the coding coefficients so that they have small within-class scatter but big between-class scatter.

$$\mathcal{L}(\boldsymbol{A}) = \sum_{c=1}^{C} \left(\sum_{y_i=c, y_j=c} \left(\frac{1}{n_c} - \frac{1}{2n}\right) \|\boldsymbol{a}_{\cdot i} - \boldsymbol{a}_{\cdot j}\|_2^2 + \sum_{y_i=c, y_j \neq c} -\frac{1}{2n} \|\boldsymbol{a}_{\cdot i} - \boldsymbol{a}_{\cdot j}\|_2^2 \right) \tag{3}$$

In Eq. (3), $y_i = c$ means the label is class c, otherwise, the label is not class c. It can be seen from this formula, the weight coefficient between the same class and different class is relatively fixed, so as to increase the discriminating ability and reduce the computational complexity. Since the coefficient of the interclass term in Eq. (3) is negative, it cannot be proved that the objective function is convex. In the process of tracking, the basis vector of the target dictionary is constantly updated, and the context relation of the basis vector of the background dictionary is required to be as weak as possible, and the new background template is also used to update. Therefore, the influence on the dictionary discriminative will be small when the class terms are removed, and the objective function can be guaranteed to be a convex function for solving. Further, the discriminant term in Eq. (3) is simplified as

$$\mathcal{L}(\boldsymbol{A},\boldsymbol{W}) = \left\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}\right\|_1 \tag{4}$$

In Eq. (4), $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_o & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_b \end{bmatrix}$, $\boldsymbol{W}_o$ is the weight coefficient matrix of target and $\boldsymbol{W}_b$ is the weight coefficient matrix of background, both can be calculated by using Eq. (2). Here, we only give the calculation of $\boldsymbol{W}_o$:

$$\boldsymbol{W}_o = \begin{bmatrix} 2 + \frac{n_0}{n} - \frac{4}{n_0} & \cdots & \frac{1}{n} - \frac{2}{n_0} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} - \frac{2}{n_0} & \cdots & 2 + \frac{n_0}{n} - \frac{4}{n_0} \end{bmatrix} \tag{5}$$

In Eq. (5), n_0 represents the number of samples of this class, and n represents the total number of samples. In conclusion, the objective function of the online non-negative discriminative dictionary learning tracking model is written again as

$$\min_{\boldsymbol{D},\boldsymbol{A}} f(\boldsymbol{D},\boldsymbol{A}) = \sum_i \sum_j \ell_\delta(x_{ij} - \boldsymbol{d}_{\cdot i}\boldsymbol{a}_{\cdot j}) + \gamma \sum_c \|\boldsymbol{A}_c\|_{1,\infty} + \frac{\beta}{2}\left\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}\right\|_1 \quad s.t. \quad \boldsymbol{D} \geq 0, \boldsymbol{A} \geq 0, \boldsymbol{d}_{\cdot k}^{\mathrm{T}}\boldsymbol{d}_{\cdot k} \leq 1, \forall \mathrm{k} \tag{6}$$

3.2 The solution of the expression coefficient *A*

After obtaining the dictionary template $\boldsymbol{D}$, there are two expression coefficients to be solved. First, the expression coefficients of candidate particles need to be solved for finding the target by the relevant generation function or discriminant function. Second, the corresponding class sparse coefficient should be solved when the template dictionary online is updated. In this section, we mainly introduce the solution method of the corresponding class sparse coefficient. Here, the objective function in Eq. (6) is a convex function with constraint term ($\boldsymbol{A} \geqslant 0$). The constraint term is written into the objective function as shown in Eq. (7) by using Lagrange multiplier method.

$$\langle \boldsymbol{A} \rangle = \min_{\boldsymbol{A}} \sum_i \sum_j \ell_\delta(x_{ij} - \boldsymbol{d}_{\cdot i}\boldsymbol{a}_{\cdot j}) + \gamma \sum_c \|\boldsymbol{A}_c\boldsymbol{A}_c\|_{1,\infty} + tr(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{A}) + \frac{\beta}{2}\left\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}\right\|_1 \tag{7}$$

Among them, the $tr(\cdot)$ is matrix rank, $\boldsymbol{\Phi}$ is the Lagrange multiplier. Due to the existence of the $\ell_{1,\infty}$ norm, the above equation is a non-smooth convex function and has not a closed solution. For the solution of $\ell_{1,\infty}$ norm, other parts must be smooth convex function. At this point, we introduce the separation variable $\boldsymbol{A}'$ and divide the solution into solving two unknown approximate functions. Then, the objective function about $\boldsymbol{A}$ is rewritten as Eq. (8):

$$\langle A, A' \rangle = \min_{A, A'} \sum_i \sum_j \ell_\delta (x_{ij} - \boldsymbol{d}_{\cdot i} \boldsymbol{a}_{\cdot j}) + \gamma \sum_c \left\| A'_c \right\|_{1,\infty} + tr(\boldsymbol{\Phi}^{\mathrm{T}} A) + \frac{\alpha}{2} \left\| A - A' \right\|_2^2 + \frac{\beta}{2} \left\| A^{\mathrm{T}} W A \right\|_1 \tag{8}$$

Since there are two unknown variables in the target function, and all unknown variables cannot be solved at one time, the most similar value needs to be obtained by multiple cross iterations (ADMM) as the solution of the objective function, so the solution can be solved in two steps again.

A)For sub-problem $\boldsymbol{A}$, we can re-design it as

$$\langle A \rangle = \min_{A} \sum_i \sum_j \ell_\delta (x_{ij} - \boldsymbol{d}_{\cdot i} \boldsymbol{a}_{\cdot j}) + tr(\boldsymbol{\Phi}^{\mathrm{T}} A) + \frac{\alpha}{2} \left\| A - A' \right\|_2^2 + \frac{\beta}{2} \left\| A^{\mathrm{T}} W A \right\|_1 \tag{9}$$

For the above equation, there is no closed solution, so the following update method that satisfies the KKT condition is used to iterate the expression coefficient $\boldsymbol{A}$ until it converges.

$$a_{kj}^{p+1} = a_{kj}^{p} \frac{\left[(Z^p \odot X)^{\mathrm{T}} D \right]_{kj}}{\left[(Z^p \odot (D(A^p)^{\mathrm{T}}))^{\mathrm{T}} D + \frac{\beta}{2} A^p W \right]_{kj} + \alpha \left(a_{kj}^{p} - a'^{p+1}_{kj} \right)} \tag{10}$$

In Eq. (10), p represents the pth iteration, $\odot$ represents the element dot product between the matrix, $\boldsymbol{X}$ is the matrix of training sample, and z_{ij} of matrix $\boldsymbol{Z}$ represents the weight coefficient of the jth characteristic of particle i, matrix $\boldsymbol{Z}$ can be obtained by Eq. (11).

$$z_{ij}^{p} = \begin{cases} 1 & \left| r_{ij}^{p} \right| < \delta \\ \frac{\delta}{r_{ij}} & \text{otherwise} \end{cases} \tag{11}$$

where $r_{ij} = x_{ij} - \boldsymbol{d}_{\cdot i} \boldsymbol{a}_{\cdot j}$ is the reconstruction of residual.

B)The corresponding sub-problem $\boldsymbol{A}'$, and the objective function is as follows:

$$\langle A' \rangle = \min_{A'} \gamma \sum_c \left(\left\| A'_c \right\|_{1,\infty} + \frac{\alpha}{2} \left\| A_c - A'_c \right\|_2^2 \right) \tag{12}$$

3.3 Dictionary template update

In the process of target tracking, in order to catch the change of target appearance in time, the new target samples are used to update the appearance presentation model constantly. In this section, it is mainly to realize the dictionary template update of the target object. Assuming that at frame l, the algorithm has obtained the position and size of the target. The target in this frame will be taken as new training samples and the corresponding dictionary will be updated. This is different from the dictionary learning method and is similar to the online dictionary learning algorithm of Mairal et al. [49] and the online non-negative dictionary learning tracking algorithm of Wang et al. [4]. Here, we adopt the dictionary updating method of Wang et al. [4].

Table 1 Evaluation results of ONDDLT with/without Fisher weight coefficient and Huber loss on OTB100 dataset

Tracker	OTB100	FPS
ONDDLT	0.418	34.6
ONDDLT without Fisher weight coefficient	0.409	18
ONDDLT without Huber loss	0.412	26.4
ONDDLT without Huber loss and Fisher weight coefficient	0.405	9

Generally speaking, in the process of target tracking, the probability of drastic change is very small, so the target between every two consecutive frames is very similar. Therefore, the training samples can be approximately divided into low-rank components and sparse components. The sparse components represent occlusion or other changes. In this way, the dictionary can automatically reduce the effect of occlusion when it is updated. Here, the algorithm still uses Eq. (6) as the target function. The optimization problem of Eq. (6) is divided into two parts: the expression coefficient $\boldsymbol{A}$ and the dictionary template $\boldsymbol{D}$. The solution of the expression coefficient $\boldsymbol{A}$ was given above. For the optimization of dictionary template $\boldsymbol{D}$, although it can be updated incrementally with a limited batch size, it needs to be completely recalculated while the new images are being inputted. Here, the mapping gradient descent method is used to solve this optimization problem as shown below:

$$\widetilde{\boldsymbol{d}}_{\cdot i}^{t} = \boldsymbol{d}_{\cdot i}^{t} - \eta \nabla h(\boldsymbol{d}_{\cdot i}^{t}), \boldsymbol{d}_{\cdot k}^{t+1} = \prod (\widetilde{\boldsymbol{d}}_{k}^{l}) \tag{13}$$

In Eq. (13), $\nabla h(\boldsymbol{d}_{\cdot i}^{t})$ is the gradient vector, and η is the update stride. The gradient vector $\nabla h(\boldsymbol{d}_{\cdot i}; \boldsymbol{A})$ corresponding to each dictionary atom $\boldsymbol{d}_{\cdot i}$ is shown as follows:

$$\nabla h(\boldsymbol{d}_{\cdot i}) = \frac{\partial h(\boldsymbol{d}_{\cdot i}; \boldsymbol{A})}{\partial \boldsymbol{d}_{\cdot i}} = A^{\mathrm{T}} \boldsymbol{\Lambda}_i A \boldsymbol{d}_{\cdot i} - A^{\mathrm{T}} \boldsymbol{\Lambda}_i y_i \tag{14}$$

where $\boldsymbol{\Lambda}_i$ denotes the diagonal matrix whose elements is the ith row in $\boldsymbol{W}^t$. $\prod(\boldsymbol{x})$ is an mapping calculation that each column element of $\boldsymbol{D}$ is mapped to the convex set $\mathcal{C} = \{\boldsymbol{x} : \boldsymbol{x} \geq 0, \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} \leq 1\}$. While solving the problem that the dictionary atom cannot be negative, it also avoids the problem of atomic scalability. Inspired by Mairal et al. [49]'s online matrix decomposition and dictionary learning, the two matrices of Eq. (13) are taken as

Table 2 Comparison in terms of expected average overlap (EAO), accuracy (A), and frames per second (FPS) on VOT2016

Tracker	ONDDLT	ONNDL	CT	MIL	TLD	L1T
EAO	0.178	0.162	0.140	0.149	0.158	0.167
A	0.52	0.49	0.42	0.42	0.44	0.50
FPS	37	7	40	10	9	0.4

The top one results are marked in red

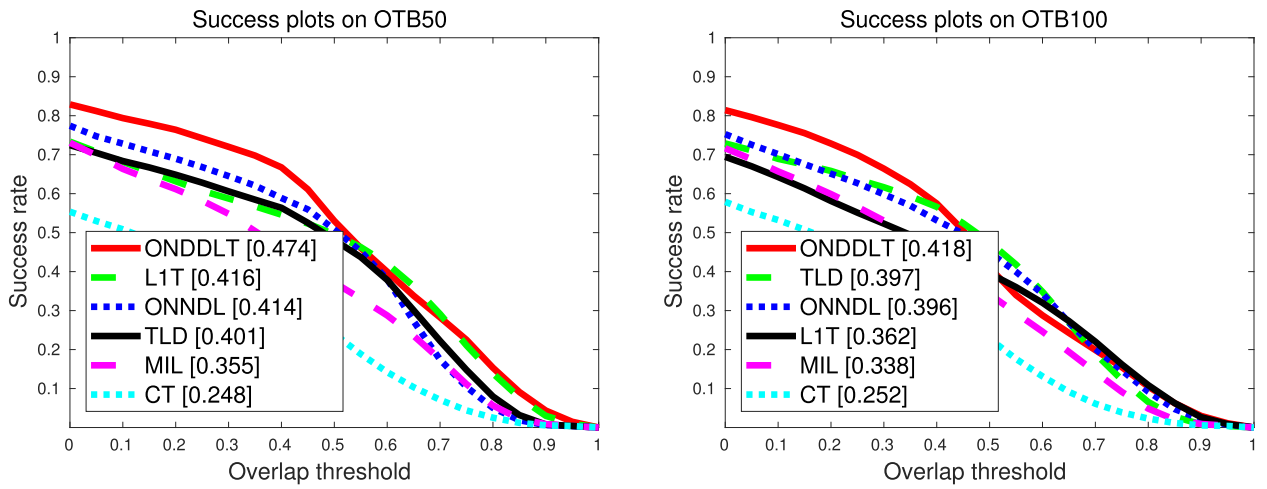

Fig. 1 Success plots compared with the relative trackers on the OTB50 and OTB100 datasets

sufficient statistical information of the sample. So that the algorithm can online update. When updating the frame l, the matrix is defined as

$$\begin{aligned} \boldsymbol{U}_i^l &= (\boldsymbol{A}^l)^{\mathrm{T}} \boldsymbol{\Lambda}_i \boldsymbol{A}^l \\ \boldsymbol{V}_i^l &= (\boldsymbol{A}^l)^{\mathrm{T}} \boldsymbol{\Lambda}_i y_i \end{aligned} \tag{15}$$

After obtaining the result of frame $l+1$, the update rules of matrix $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are

$$\begin{aligned} \boldsymbol{U}_i^{l+1} &= \rho \boldsymbol{U}_i^l + \boldsymbol{a}_{\cdot l+1} \boldsymbol{a}_{\cdot l+1}^{\boldsymbol{T}} \\ \boldsymbol{V}_i^{l+1} &= \rho \boldsymbol{V}_i^l + \boldsymbol{a}_{\cdot l+1} y_i \end{aligned} \tag{16}$$

In the formula, the ρ is the forgetting factor. It is the exponential reduction of previous data. In summary, the atomic update rules of the dictionary template are as follows:

$$\tilde{\boldsymbol{d}}_{\cdot i}^{\,l} = \boldsymbol{d}_{\cdot i}^{\,l} - \eta \left(\boldsymbol{U}_i^{l+1} \boldsymbol{d}_{\cdot i}^{\,l+1} - \boldsymbol{V}_i^{l+1} \right), \boldsymbol{d}_{\cdot k}^{\,l+1} = \prod (\tilde{\boldsymbol{d}}_{\cdot k}^{\,l}) \tag{17}$$

3.4 Target positioning module

When locating the target, the feature extraction and selection of the target image are needed first. Generally, rectangular bounding box is used as the size and position of the target. However, the target is not always rectangular, so even in the correct target image blocks or the real target location and size (ground truth), it is inevitable to contain a small number of background areas. In addition, the deformation or occlusion of target will have adverse effects for tracking. These effects can be reduced if the invariant feature and informational characteristics of the target are selected. For this reason, the feature selection by logistic regression with ℓ_1 norm is adopted in this section as shown below:

$$\min_{\boldsymbol{w}} \sum_i \log\{1 + \exp[-l_i(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{y}_i + b)]\} + \xi \|\boldsymbol{w}\|_1 \tag{18}$$

where $\boldsymbol{y}_i$ is a sample in the previous frame, and l_i is the corresponding label. When the value is 1, it indicates that

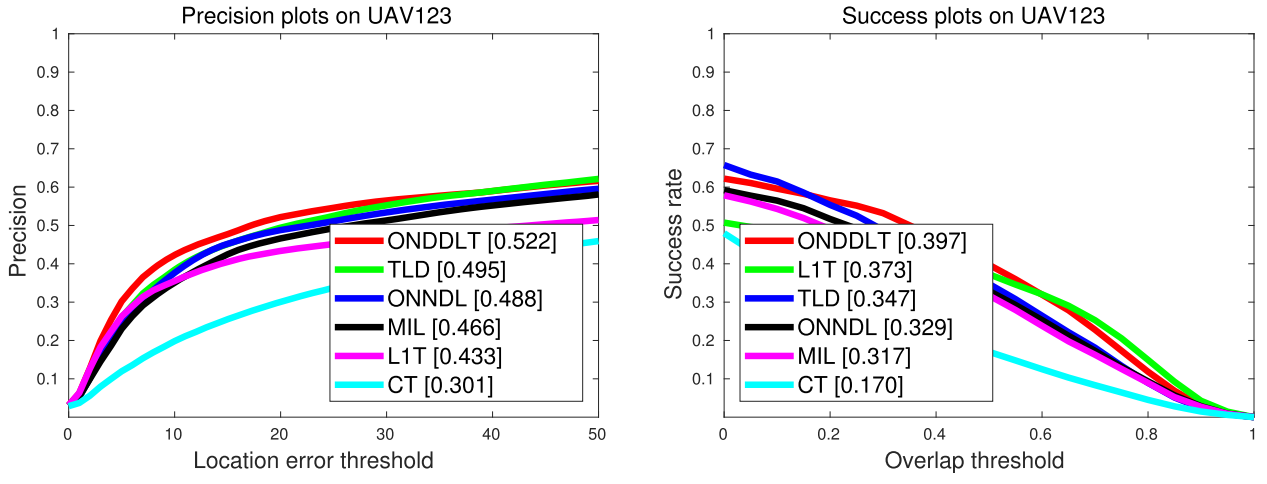

Fig. 2 Success plots of IPR, IV, LR, and BC attributes on the OTB100 dataset

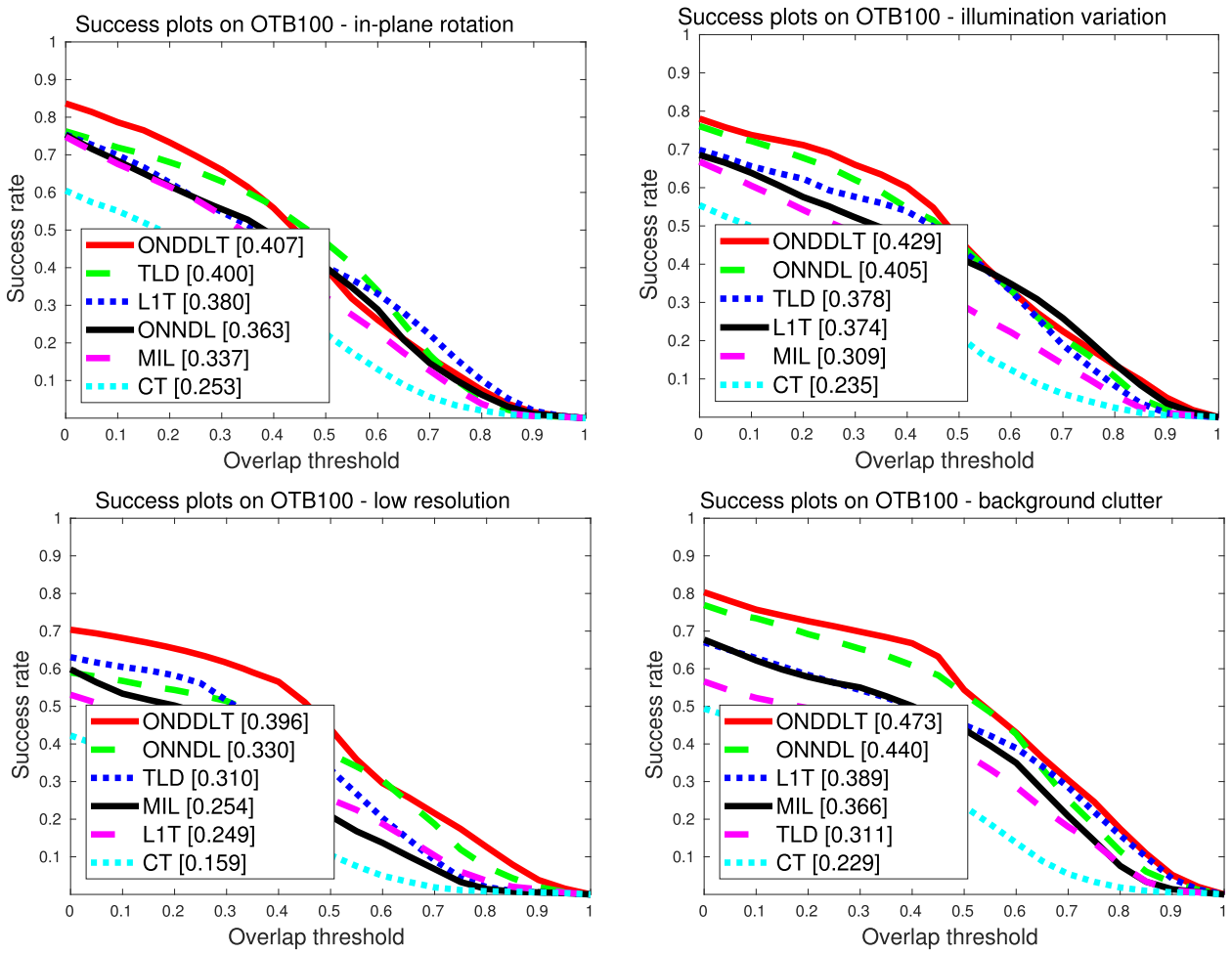

Fig. 3 Precision plots and success plots compared with the relative trackers on the UAV123 dataset

$\boldsymbol{y}_i$ is a positive sample. When the value is -1, it indicates that $\boldsymbol{y}_i$ is a negative sample. By feature selection, the computational complexity of the algorithm is reduced while the robust robust discriminative samples are provided. In a series of tracking algorithm studies, it is shown that detailed grid search is not suitable for most algorithms. Because that high similarity between samples leads to redundancy and the redundancy and the computational complexity increases with the square multiple of the target image size. Therefore, we use the particle filter based on the sequence monte carlo (SMC) model to select the samples. Particle filter is a kind of sample selection method and is used frequently in visual tracking due to its simplicity and high efficiency. The particle filter dynamically provides a candidate sample for the tracking algorithm by estimating the hidden state sequentially by observing the sequence. The hidden state variable of particle filter does not need to strictly follow Gaussian distribution or some distribution with parameters. At the same time, with the increase of the number of filters, the approximation precision also increases. In addition, the probability distribution of the hidden state variables can make algorithm easier to recover from tracking failure.

Table 3 Comparison results of the AUC score (%) on OTB50 dataset with different values of parameters α, β, and η

Parameters	$\beta = 0.005$			$\beta = 0.01$		
	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.15$
$\eta = 0.1$	46.8	46.6	45.8	45.4	45.2	44.5
$\eta = 0.2$	47.4	47.1	46.4	46.4	46.5	46.1
$\eta = 0.3$	47.0	46.4	45.8	46.6	46.4	45.9

The objective function of Eq. (6) is adopted as the search method when we choose the suitable target from the candidate particles. When the target is positioned, the particle representation coefficient is independent existence rather than a certain class or group. Therefore, the regularization term of the target function cannot be applied to encode the coefficient of a class or group with the $\ell_{1,\infty}$ norm. Here, the regularization term of the target

function can be redefined as Eq. (18) after adopting ℓ_1 norm.

$$\min_{A} \sum_{i} \sum_{j} \ell_{\delta}(x_{ij} - \boldsymbol{d}_{\cdot i}\boldsymbol{a}_{\cdot j}) + \gamma \|\boldsymbol{A}\|_1 + tr(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{A}) + \frac{\beta}{2}\left\|\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}\right\|_1 \tag{19}$$

Similar to the Eq. (10), there is no closed solution in the above equation, but the expression coefficient $\boldsymbol{A}$ can be iterated until convergence by using the following update method which satisfies the KKT condition:

$$a_{kj}^{p+1} = a_{kj}^{p} \frac{\left[(\boldsymbol{Z}^{p} \odot \boldsymbol{X})^{\mathrm{T}}\boldsymbol{D}\right]_{kj}}{\left[(\boldsymbol{Z}^{p} \odot (\boldsymbol{D}(\boldsymbol{A}^{p})^{\mathrm{T}}))^{\mathrm{T}}\boldsymbol{D} + \frac{\beta}{2}\boldsymbol{A}^{p}\boldsymbol{W}\right]_{kj}} \tag{20}$$

After obtaining the representation coefficient, the particle with the maximum reconstruction value in the target dictionary template is usually used as the predicted value, but this method is easy to cause the problem of target drift. The target dictionary template and the background dictionary template are combined to improve the robustness of the algorithm. That is, the reconstruction value of the target should be as large as possible and the reconstruction value of the background should be as small as possible. Thus, $\mu(\|\boldsymbol{D}_o\boldsymbol{a}_o\|_1 - \|\boldsymbol{D}_b\boldsymbol{a}_b\|_1)$ is used as the target function, the subscript o, b respectively represents the target and background. The parameter μ is mainly used to control the sparse representation of particles and constraint representation of background.

The entire steps of our ONDDLT are summarized in Algorithm (1).

Algorithm 1: Online non-negative discriminative dictionary learning for tracking

Input: The initial bounding box $\boldsymbol{b_0}$

Output: The predicted target state $\boldsymbol{b}_l = (\hat{x}_{l+1}, \hat{y}_{l+1}, \hat{s}_{l+1})$, dictionary template $\boldsymbol{D}$ and expression coefficient $\boldsymbol{A}$

repeat

1.Getting the feature of frame $l+1$ by solving the logistic regression function

2.Solving the candidate samples coefficient

while $\left\|A^{p+1} - A^{p}\right\|_2 < \varepsilon_1$ **do**

for *the elements* a_{kj}^{p+1} *in* $\boldsymbol{A}$ **do**

Solving the Eq. (20) to obtain the elements a_{kj}^{p+1} in candidate samples coefficient $\boldsymbol{A}$

end

end

3.Predicting the position and scale of target $(\hat{x}_{l+1}, \hat{y}_{l+1}, \hat{s}_{l+1}) \leftarrow \max(\boldsymbol{D}_0\boldsymbol{a}_0 - \boldsymbol{D}_b\boldsymbol{a}_b)$

4.Solving the training samples coefficient

while $\left\|A^{t+1} - A^{t}\right\|_2 < \varepsilon_0$ **do**

while $\left\|A^{p+1} - A^{p}\right\|_2 < \varepsilon_1$ **do**

for *the elements* a_{kj}^{p+1} *in* $\boldsymbol{A}$ **do**

Solving the Eq. (10) to obtain the elements a_{kj}^{p+1} in training samples coefficient $\boldsymbol{A}$

end

end

Solving the Eq. (12) to obtain the $\boldsymbol{A}'$

end

5.Update dictionary template

for $i = 1$ *to* n **do**

Update dictionary template by solving the Eq. (13)

end

until *The end of the video sequences*;

4 Experimental results and discussion

4.1 Visual tracker benchmark

In this section, our trackers is evaluated on OTB50 [50] and OTB100 [51] datasets. The OTB50 dataset with 50 fully annotated sequences is to facilitate tracking evaluation. In order to increase the robustness of evaluation, the OTB100 dataset adds 50 videos compared with OTB50 dataset. For further analysis, the dataset labeled every video with 11 attributes(illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, low resolution). We use the one pass evaluation (OPE) with success plot for evaluation which counts the number of successful frames whose overlap are larger than the given threshold. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. To verified the robustness of the tracker, we also performed tracker on VOT2016 and UAV123 datasets. The VOT2016 dataset contains 60 sequences and the UAV123 dataset contains 123 sequences. The VOT2016 benchmark introduced the expected average overlap (EAO) to measure the expected no-reset overlap of a tracker. The videos in UAV123 dataset were captured from low-altitude UAVs. Evaluation on the UAV123 dataset is to better measure the performance of the tracker in different scenarios.

4.2 Ablation study

For an in depth analysis of the Fisher weight coefficient and Huber loss, we evaluate each component on the OTB100 dataset respectively. As can be seen from Table 1, the tracking speed are also improved by Fisher weight coefficient and Huber loss. Removing the effect of Fisher weight coefficient causes the FPS from 34.6 to 18 with a decrease of half and the AUC scores decreases about 0.9%. The Huber loss also improves the accuracy and the speed. Overall, the ablation study results demonstrate the effectiveness of the Fisher weight coefficient and Huber loss for tracking task.

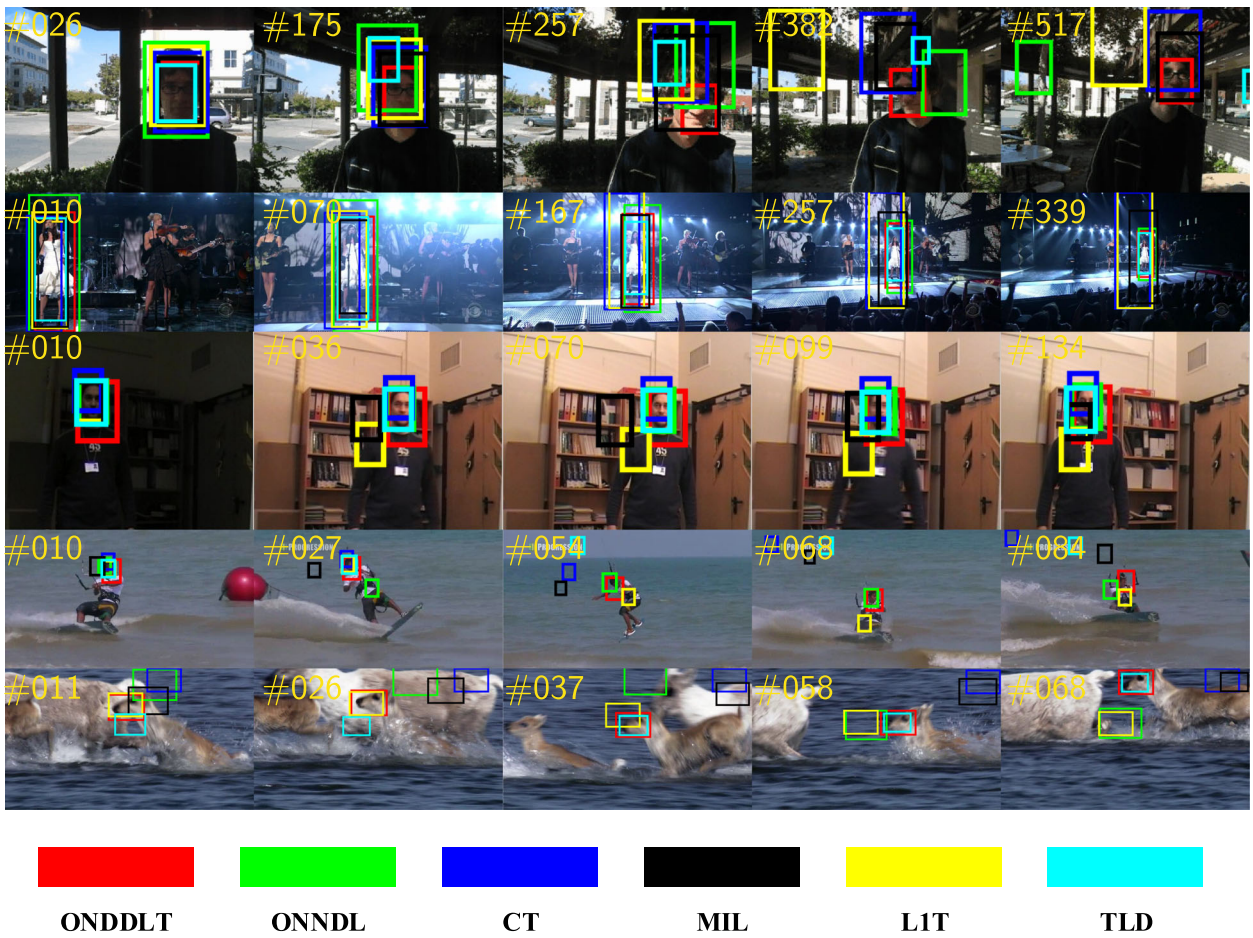

Fig. 4 Visualization performance of our method compared with other relative trackers on Trellis, Singer, Man, Kitesurf, and Deer

4.3 Quantitative analysis

To validate the proposed method, our tracker is compared with the relative trackers, including ONNDL [4], CT [14], MIL [21], L1T [11], and TLD [19]. In the evaluation, we set $\alpha = 0.05, \delta = \gamma = \xi = 0.01, \rho = 0.99, \eta = 0.2$, and $\beta = 0.005$. We compared the tracking speed of trackers in Table 2 on VOT2016 dataset, our ONDDLT can achieve real-time tracking while improving the precision.

Figure 1 presents the comparison results on OTB50 and OTB100 datasets. Compared to the relative ONNDL and L1T, the performance of ONDDLT has improved a lot on OTB50 dataset, achieving the best success rate of 47.4% and improving the precision by 6.0% and 5.8%, respectively. We notice that the performance of ONDDLT has declined a lot on OTB100 dataset. However, our tracker also achieves the best performance (41.8%) on OTB100 dataset. We find that our tracker gets superior performance than ONNDL with a gain of 2.2%. Overall, our ONDDLT performs excellent against other relative trackers on public visual tracking benchmarks. For comprehensive analysis, the success plot over different video attributes annotated is presented in the OTB100 benchmark. On average, our ONDDLT performs better about 3% higher than ONNDL on all attributes. Our method is ranked top 1 on 8 attributes and top 2 on 2 attributes, which can be explained by the advantages of the global dictionary learning model and the class-specific dictionary learning model. Especially, our tracker obtains significant improvements on LR, IV, BC, and IPR shown as in Fig. 2. More results can be found in Figure 5 of Appendix A. According to the comparison results in Table 2 and Fig. 3, the performance of our ONDDLT are all top one in VOT2016 and UAV123 datasets. The accuracy of ONNDL decreases rapidly in UAV123 dataset, but our ONDDLT shows good robustness. To verify parameter robustness of trackers, we selected three important parameters that affect the tracking accuracy in Table 3. We can see that the accuracy of the tracker remains stable within a certain range of parameters. When $\alpha = 0.05, \beta = 0.005$. and $\eta = 0.2$, the AUC score achieves optimal accuracy.

4.4 Qualitative analysis

The qualitative results are presented by visualization in Fig. 4. These sequences are captured under the conditions

of complicated environment. The L1T loses the target on all videos and the ONNDL loses the target on Trellis, Kitesurf, and Deer. We can know that ONDDLT has stronger robustness and higher accuracy compared with the tracker based on sparse coding. As you can see from the picture, our tracker predicts more accurate position and scales than the other methods. Specifically, our method not only has better robustness and accuracy for light variation on Trellis and Man, but also has better robustness and accuracy for scale variation on Singer. Our method also performs well on fast motion (Deer) and deformation (Kitesurf) while other trackers lost the target. The reason of improvement of tracking performance is that online non-negative discriminant dictionary learning tracking strategy is used to improve the discriminative ability for matching.

5 Conclusion

In this paper, online non-negative discriminative dictionary learning for tracking algorithm is proposed, which combines the advantages of the global dictionary learning model and the class-specific dictionary learning model. To this end, we explore online dictionary learning tracking algorithm and introduce the online discriminant dictionary learning tracking strategy. Especially, the Huber loss function and the Fisher weight coefficient is used in the discriminative term to improve computational efficiency. In addition, non-negative constraints on dictionaries is added to enhance the performance. The experimental results show that our method performs much better than the tracking method compared in this paper. Compared with current shallow features, deep learning can more adaptively explore the semantic features of the target. Therefore, the fusion of deep learning and sparse representation can be studied. In addition, the computational efficiency and performance of the tracking algorithm based on sparse coding can be further optimized.

Appendix: Evaluation results of different attributes on the OTB100 dataset

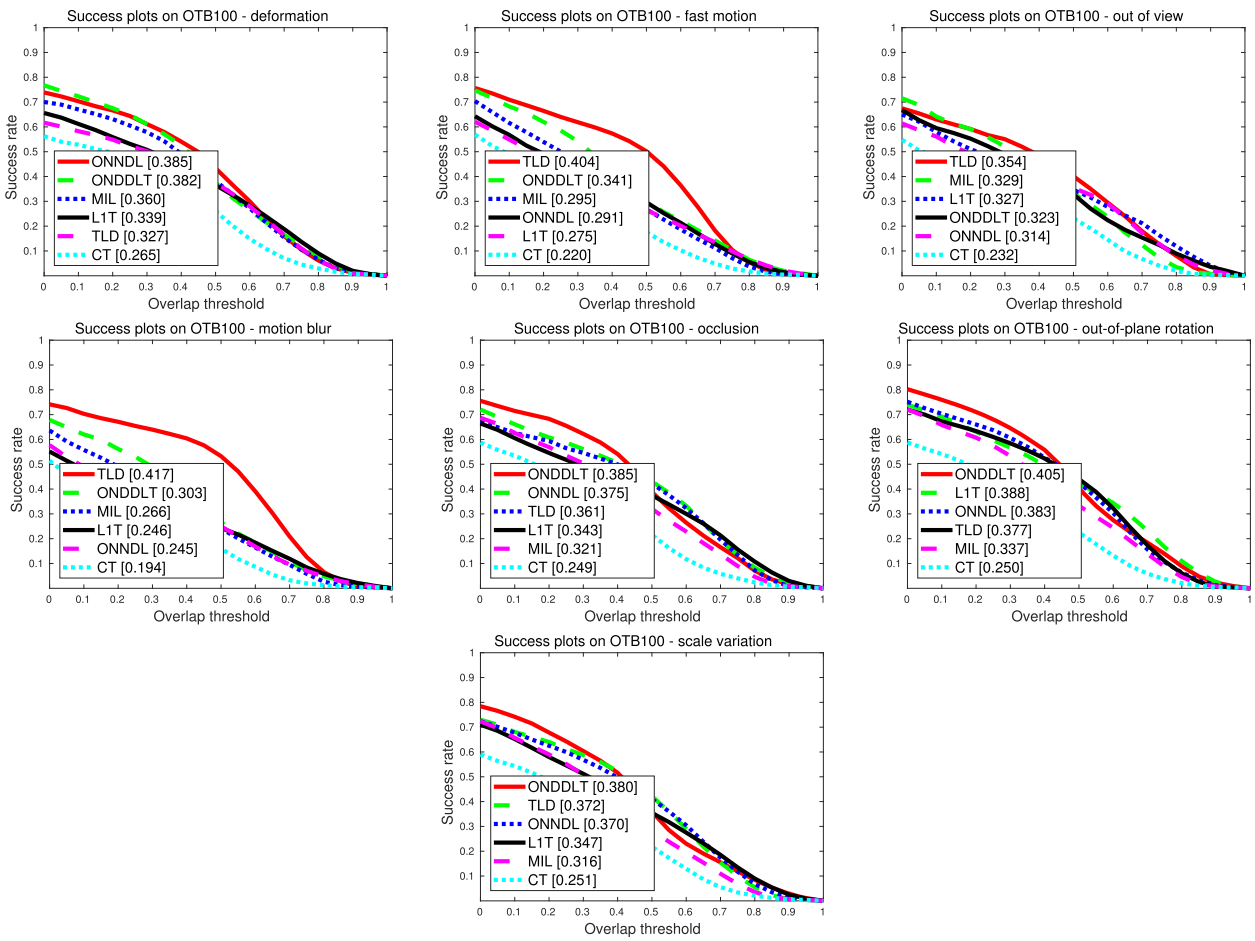

Fig. 5 Success plots of DEF, FM, OV, MB, OCC, OPR, and SV attributes on the OTB100 dataset

Abbreviations
ADMM: Multiple cross iterations; BC: Background clutters; CF: Correlation filter; CT: Compressive tracking; DEF: Deformation; DL: Deep leaning; FM: Fast motion; IPR: In-plane rotation; IV: Illumination variation; IVT: Incremental visual tracking; LR: Low resolution; MB: Motion blur; OCC: Occlusion; ONDDLT: Online non-negative discriminative dictionary learning for tracking; ONNDL: Online non-negative dictionary learning target tracking method; OPE: One pass evaluation; OPR: Out-of-plane rotation; OV: Out-of-view; PCA: Principal component analysis; SMC: Sequence Monte Carlo; SV: Scale variation; SVM: Support vector machine; TLD: Tracking learning detection; VTD: Visual tracking decomposition

Acknowledgements
This work was supported by a grant from the National Natural Science Foundation of China (NSFC) of Grant Nos. 61502275, Natural Science Foundation of Shandong Province (No. ZR2019MF011) and Project funded by China Postdoctoral Science Foundation (No. 2017M622210). This work was also supported in part by National Natural Science Foundation of China-Yunnan Joint Fund (61872118). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V used for this research.

Authors' contributions
WW and PL conceived the algorithm and designed the experiments. WW performed the experiments. PL and FY analyzed the results. WW drafted the manuscript. FY, HZ, and WZ revised the manuscript. All authors read and approved the final manuscript.

Funding
The work was supported by the National Natural Science Foundation of China (NSFC) (grant nos. 61502275 and 61872118) and the Postdoctoral Science Foundation of China (grant no. 2017M622210).

Availability of data and materials
Please contact the corresponding author for data requests.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Shandong University at Weihai, School of Mechanical, Electrical and Information Engineering, Wenhua Xilu, 264209 Weihai, China. [2]Harbin Institute of Technology, Vision Perception and Cognition Lab, School of Computer Science and Technology, West Street, 150001 Harbin, China.

Received: 9 March 2019 Accepted: 23 August 2019
Published online: 30 October 2019

References

1. W. Zhang, K. Song, X. Rong, Y. Li, Coarse-to-fine uav target tracking with deep reinforcement learning. IEEE Trans. Autom. Sci. Eng., 1–9 (2018). https://doi.org/10.1109/tase.2018.2877499
2. K. Song, W. Zhang, X. Rong, in *2018 24th International Conference on Pattern Recognition (ICPR)*. Uav target tracking with a boundary-decision network (IEEE, 2018), pp. 2576–2581. https://doi.org/10.1109/icpr.2018.8545872
3. A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: An experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
4. N. Wang, J. Wang, D. Y. Yeung, in *International Conference on Computer Vision*. Online robust non-negative dictionary learning for visual tracking (IEEE, 2013). https://doi.org/10.1109/iccv.2013.87
5. T. Liu, W. Gang, Q. Yang, in *IEEE Conference on Computer Vision & Pattern Recognition*. Real-time part-based visual tracking via adaptive correlation filters (IEEE, 2015). https://doi.org/10.1109/cvpr.2015.7299124
6. N. Wang, S. Li, A. Gupta, D.-Y. Yeung, Transferring rich feature hierarchies for robust visual tracking. arXiv preprint (2015). arXiv:1501.04587
7. S. Hong, T. You, S. Kwak, B. Han, in *32nd International Conference on Machine Learning, ICML 2015*. Online tracking by learning discriminative saliency map with convolutional neural network, vol. 1, (Lile, 2015), pp. 597–606
8. D. A. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008)
9. B. Han, D. Comaniciu, Y. Zhu, L. S. Davis, Sequential kernel density approximation and its application to real-time visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **30**(7), 1186–1197 (2008)
10. J. Kwon, K. M. Lee, in *IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking decomposition (IEEE, 2010), pp. 1269–1276. https://doi.org/10.1109/cvpr.2010.5539821
11. X. Mei, H. Ling, in *International Conference on Computer Vision*. Robust visual tracking using ℓ_1 minimization (IEEE, 2009), pp. 1436–1443. https://doi.org/10.1109/iccv.2009.5459292
12. X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, in *IEEE Conference on Computer Vision & Pattern Recognition*. Minimum error bounded efficient ℓ_1 tracker with occlusion detection (IEEE, 2011), pp. 1257–1264. https://doi.org/10.1109/cvpr.2011.5995421
13. W. Zhong, H. Lu, M.-H. Yang, in *IEEE Conference on Computer Vision & Pattern Recognition*. Robust object tracking via sparsity-based collaborative model (IEEE, 2012), pp. 1838–1845. https://doi.org/10.1109/cvpr.2012.6247882
14. K. Zhang, L. Zhang, M.-H. Yang, in *European Conference on Computer Vision*. Real-time compressive tracking, (2012), pp. 864–877
15. L. Cehovin, M. Kristan, A. Leonardis, Robust visual tracking using an adaptive coupled-layer visual model. IEEE Trans. Pattern Anal. Mach. Intell. **35**(4), 941–953 (2013)
16. J. T. Zhou, I. W. Tsang, S. J. Pan, M. Tan, Multi-class heterogeneous domain adaptation. J. Mach. Learn. Res. **20**(57), 1–31 (2019)
17. B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
18. Z. Kalal, J. Matas, K. Mikolajczyk, in *IEEE Conference on Computer Vision & Pattern Recognition*. Pn learning: Bootstrapping binary classifiers by structural constraints (IEEE, 2010), pp. 49–56. https://doi.org/10.1109/cvpr.2010.5540231
19. Z. Kalal, K. Mikolajczyk, J. Matas, et al., Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409 (2012)
20. S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, P. H. Torr, Struck: Structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
21. K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning. Pattern Recogn. **46**(1), 397–411 (2013)
22. D. S. Bolme, B. A. Draper, J. R. Beveridge, in *IEEE Conference on Computer Vision & Pattern Recognition*. Average of synthetic exact filters (IEEE, 2009), pp. 2105–2112. https://doi.org/10.1109/cvprw.2009.5206701
23. D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, in *IEEE Conference on Computer Vision and Pattern Recognition*. Visual object tracking using adaptive correlation filters (IEEE, 2010), pp. 2544–2550. https://doi.org/10.1109/cvpr.2010.5539960
24. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, in *European Conference on Computer Vision*. Exploiting the circulant structure of tracking-by-detection with kernels (Springer Berlin Heidelberg, 2012), pp. 702–715. https://doi.org/10.1007/978-3-642-33765-9_50
25. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
26. Y. Yao, X. Wu, L. Zhang, S. Shan, W. Zuo, in *Proceedings of the European Conference on Computer Vision (ECCV)*. Joint representation and truncated inference learning for correlation filter based tracking (Springer International Publishing, 2018), pp. 552–567. https://doi.org/10.1007/978-3-030-01240-3_34
27. F. Yang, Z. Jiang, L. S. Davis, in *IEEE Winter Conference on Applications of Computer Vision*. Online discriminative dictionary learning for visual tracking (IEEE, 2014), pp. 854–861. https://doi.org/10.1109/wacv.2014.6836014
28. T. Zhang, B. Ghanem, S. Liu, N. Ahuja, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Low-rank sparse learning for robust visual tracking, vol. 7577 LNCS, PART 6, (Florence, 2012), pp. 470–484. http://dx.doi.org/10.1007/978-3-642-33783-3_34
29. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)

30. Y. Lu, T. Wu, S. Chun Zhu, in *IEEE Conference on Computer Vision & Pattern Recognition*. Online object tracking, learning and parsing with and-or graphs (IEEE, 2014), pp. 3462–3469. https://doi.org/10.1109/cvpr.2014.443
31. Y. Jiang, J. Ma, in *IEEE Conference on Computer Vision & Pattern Recognition*. Combination features and models for human detection (IEEE, 2015), pp. 240–248. https://doi.org/10.1109/cvpr.2015.7298620
32. S. Zhang, Y. Qi, F. Jiang, X. Lan, P. C. Yuen, H. Zhou, Point-to-set distance metric learning on deep representations for visual tracking. IEEE Trans. Intell. Transp. Syst. **19**(1), 187–198 (2018)
33. Y. Qi, S. Zhang, L. Qin, Q. Huang, H. Yao, J. Lim, M.-H. Yang, Hedging deep features for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **41**(5), 1116–1130 (2018)
34. J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, R. S. M. Goh, Transfer hashing: From shallow to deep. IEEE Trans. Neural Netw. Learn. Syst. **29**(12), 6191–6201 (2018)
35. J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, R. S. M. Goh, Anomalynet: An anomaly detection network for video surveillance. IEEE Trans. Inf. Forensic Secur. **14**(10), 2537–2550 (2019). https://doi.org/10.1109/tifs.2019.2900907
36. L. Qu, K. Liu, B. Yao, J. Tang, W. Zhang, Real-time visual tracking with elm augmented adaptive correlation filter. Pattern Recogn. Lett. (2018). https://doi.org/10.1016/j.patrec.2018.09.015
37. J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning. IEEE Trans. Pattern Anal. Mach. Intell., 791–804 (2012)
38. D. Pham, S. Venkatesh, in *IEEE Conference on Computer Vision & Pattern Recognition*. Joint learning and dictionary construction for pattern recognition (IEEE, 2008), pp. 1–8. https://doi.org/10.1109/cvpr.2008.4587408
39. S. Cai, W. Zuo, L. Zhang, X. Feng, P. Wang, in *European Conference on Computer Vision*, ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Support vector guided dictionary learning (Springer International Publishing, 2014), pp. 624–639. https://doi.org/10.1007/978-3-319-10593-2_41
40. I. Ramirez, P. Sprechmann, G. Sapiro, in *IEEE Conference on Computer Vision & Pattern Recognition*. Classification and clustering via dictionary learning with structured incoherence and shared features (IEEE, 2010), pp. 3501–3508. https://doi.org/10.1109/cvpr.2010.5539964
41. X. Lan, S. Zhang, P. C. Yuen, in *IJCAI International Joint Conference on Artificial Intelligence*. Robust joint discriminative feature learning for visual tracking, vol. 2016, (New York, 2016), pp. 3403–3410
42. X. Lan, M. Ye, S. Zhang, P. C. Yuen, in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. Robust collaborative discriminative learning for RGB-infrared tracking, (New Orleans, 2018)
43. X. Lan, M. Ye, R. Shao, B. Zhong, P. C. Yuen, H. Zhou, Learning modality-consistency feature templates: a robust rgb-infrared tracking system. IEEE Trans. Ind. Electron. **66**(12), 9887–9897 (2019)
44. M. Yang, L. Zhang, X. Feng, D. Zhang, in *International Conference on Computer Vision*. Fisher discrimination dictionary learning for sparse representation (IEEE, 2011), pp. 543–550. https://doi.org/10.1109/iccv.2011.6126286
45. X. Lan, P. C. Yuen, R. Chellappa, in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. Robust MIL-based feature template learning for object tracking, (2017), p. 4118
46. X. Lan, S. Zhang, P. C. Yuen, R. Chellappa, Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker. IEEE Trans. Image Process. **27**(4), 2022–2037 (2017)
47. S. Zhang, X. Lan, H. Yao, H. Zhou, D. Tao, X. Li, A biologically inspired appearance model for robust visual tracking. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2357–2370 (2016)
48. X. Lan, A. J. Ma, P. C. Yuen, R. Chellappa, Joint sparse representation and robust feature-level fusion for multi-cue visual tracking. IEEE Trans. Image Process. **24**(12), 5826–5841 (2015)
49. J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res. **11**(Jan), 19–60 (2010)
50. Y. Wu, J. Lim, M. H. Yang, in *IEEE Conference on Computer Vision & Pattern Recognition*. Online object tracking: A benchmark (IEEE, 2013), pp. 2411–2418. https://doi.org/10.1109/cvpr.2013.312
51. Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)

Publisher's Note