Wang and Xu *EURASIP Journal on Wireless Communications and Networking* (2017) 2017:194
DOI 10.1186/s13638-017-0982-4

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

# Robust L1 tracker with CNN features

Hongqing Wang and Tingfa Xu*

## Abstract

Recently, L1 tracker has been widely applied and received great success in visual tracking. However, most L1 trackers use only the image intensity for sparse representation, which is insufficient to represent the object especially when drastic appearance changes occur. Convolutional neural network (CNN) has demonstrated remarkable capability in a wide range of computer vision fields, and features extracted from different convolutional layers have different characteristics. In this paper, we propose a novel sparse representation model with convolutional features for visual tracking. Besides, to alleviate the redundancy from high-dimensional convolutional features, a feature selection method is adopted to remove noisy and irrelevant feature maps, which can reduce computation redundancy and improve tracking accuracy. Different from traditional sparse representation based tracking methods, our model not only exploits convolutional features to improve the robustness for describing the object appearance but also uses the trivial templates to model both reconstruction errors caused by sparse representation and the eigen-subspace representation. In addition, an unified objective function is proposed and a customized APG method is developed to effectively solve the optimization problem. Numerous qualitative and quantitative evaluations demonstrate that our tracker outperforms other state-of-the-art trackers in a wide range of tracking scenarios.

**Keywords:** CNN features, Visual tracking, Sparse representation, APG method

## 1 Introduction

Visual tracking plays an important role in computer vision and has received fast-growth attention in recent years due to its wide practical application such as pedestrian detection, vehicle navigation, security surveillance, and wireless communication [1–4]. In general, visual tracking is to track an interested target which usually has been indicated in the first frame by a bounding box in video streams. The main challenge of visual tracking is the numerous appearance changes, such as occlusion, abrupt motion, illumination variation, in-plane rotation, out-of-plane rotation, deformation, and scale variation.

To overcome the above challenges, many effective trackers have been proposed in recent years [5, 6]. Generally, tracking algorithms can be classified into three different categories: discriminative, generative, and hybrid generative-discriminative. Discriminative trackers formulate tracking as a binary classification problem, which search the target location and extract target from the background. The main problem of discriminative trackers is that they cannot estimate the target-specific location due to the limited number of candidates. Generative trackers adopt an appearance model to represent the target appearance, which estimate the target state by finding the highest likelihood; the model is often updated online to deal with the appearance changes. The main problem of generative trackers is that the appearance model often exhibits some limitations thus cannot represent the target effectively [7–9]. Hybrid generative-discriminative trackers fuse the advantages of discriminative and generative trackers; researchers have proposed many effective hybrid generative-discriminative trackers recently [10–13]. The hybrid model can take advantage of the global characteristics of object and also exploit the useful information from the background. However, the complexity of hybrid model is relatively high, which will lead to a

* Correspondence: xutingfa1123@163.com
School of Optoelectronics, Image Engineering & Video Technology Lab, Beijing Institute of Technology, Beijing 100081, China

high computational cost, thus cannot meet the requirement in practice.

Recently, as a generative tracking algorithm, sparse representation model has achieved outstanding performance [14]. Mei et al. [15] first propose the L1 tracker by casting the tracking problem as finding a sparse combination of a target template set and a trivial template set to approximate the target object. Then the sparsity is achieved by solving an $l_1$-regularized least squares problem. Ji et al. [16] improve the tracking accuracy by adding an $l_2$ norm regularization on the trivial coefficients and use an accelerated proximal gradient approach for solving the minimization problem which has achieved both tracking accuracy and computational efficiency. Zhang et al. [17] exploit the intrinsic relationship among different candidates which utilize the joint-sparsity property of candidates by casting tracking as a multi-task problem. Jia et al. [18] exploit both partial information and spatial information of the target based on a novel alignment-pooling method and employ a template update strategy, which combines incremental subspace learning and sparse representation. Wang et al. [19] introduce $l_1$ regularization into the principal component analysis (PCA) reconstruction and propose an online tracking algorithm, which approximates the target by linearly combining the PCA basis and a sparse set of trivial templates. Liu et al. [20] use a local sparse representation for representing the target and exploit the sparse coding histogram to represent the dynamic dictionary basis distribution of the target model. Guo et al. [21] propose a novel multi-view structural local subspace method which jointly exploits the advantages of three sub-models and uses an alignment-weighting average method to obtain the optimal state of the target. Wang et al. [22] adopt squared templates to replace trivial templates to handle partial occlusion and propose a probabilistic collaborative representation framework, which reduces the complexity in traditional sparse model-based methods. Kim et al. [23] propose a novel structure-preserving sparse learning method, which preserves both local geometrical and discriminative structures within a multi-task feature selection framework.

However, most of these methods mainly aim at improving the tracking accuracy or efficiency, they usually use the image intensity to construct the template set, which is less effective in expressing the structural information of the target, thus cannot cover severe appearance changes of the target object.

To solve this problem, many hand-crafted features have been used for visual tracking, such as Haar-like features, histogram of oriented gradient (HOG) features, local binary pattern (LBP), and scale-invariant feature transform (SIFT). However, these hand-crafted features are not robust for generic object tracking. Convolutional neural network (CNN) models which learn hierarchical features from raw images on large-scale dataset have been widely used to represent the appearance of the target. Ma et al. [24] exploit the features from hierarchical layers of CNN within a correlation filter-based framework for visual tracking, learn linear correlation filters on each CNN layer, and adopt a coarse-to-fine method to estimate the target location. Wang et al. [25] analyze CNN features from different layers and use a novel tracking method which jointly exploits two convolutional layers to mitigate the drift problem. Danelljan et al. [26] indicate that activations from the first convolutional layers achieve favorable tracking performance compared with the deeper layers within a discriminative correlation filter-based framework. In contrast to the traditional feature descriptors, CNN features contain more structural information, which is crucial to localize the target in an unknown frame.

Motivated by the above observations, we present a novel L1 tracker with CNN features. The proposed approach use a novel sparse representation model with convolutional features for visual tracking, which not only exploits CNN features to improve the robustness for describing the object appearance but also uses the trivial templates to model both reconstruction errors caused by sparse representation and the eigen-subspace representation. Besides, to alleviate redundancy of high-dimensional convolutional features, a feature selection method is adopted, which can reduce computation complexity and improve tracking accuracy. This strategy makes the model jointly exploit the advantages of the CNN features with more structural information to effectively represent the target, and of both sparse representation and the incremental subspace learning simultaneously. In addition, a customized APG method is developed to effectively solve the optimization problem. Furthermore, a robust observation likelihood metric is proposed.

The rest of this paper is organized as follows. In Section 2, we introduce the CNN features and the proposed sparse model in detail. In Section 3, we demonstrate the optimization of the objective function and the overall tracking algorithm. In Section 4, we present the details of the quantitative and qualitative experiments of our method compared with the state-of-the-art methods. In Section 5, we reach the conclusions of this paper.

## 2 Proposed model

### 2.1 CNN features

Most of the traditional L1 trackers usually use the image intensity to construct the template set. However, the image intensity-based trackers can hardly handle the complicated situation in practical visual tracking due to the lack of target structural information. To this end, our algorithm introduces CNN features in describing the target template set.

Convolutional neural network (CNN) has been successfully applied in many computer vision fields, especially in complicated tasks such as object detection, image classification, and object recognition [27]. Traditional CNN, which only the information from the last layer are used to represent the target, are effective in dealing with classification problems. However, adopting CNN for generic visual tracking directly is inadequate due to the lack of training samples and the computational complexity.

To overcome this problem, pre-trained CNN feature extraction method is proposed in recent years. CNN features, extracted from different CNN layers, have different characteristics in describing the object [24]. The CNN features from deeper layer contains more high-level semantic information, which can be seen as structural information, have more distinguishing capabilities and thus is effective facing the situation when intra-class appearance variation occurs. However, the features from deep layer have very low spatial resolution so that it cannot fit the task in generic visual tracking, which aim to indicate the location of target. On the other hand, CNN features from earlier layer contain more fine-grained information, which means the more the discriminative capabilities, the more effective in locating the target. But with the less semantic information, features from earlier layer are more sensitive to intra-class appearance variations.

From the observation above, different from the common strategy which use CNN feature extracted from the last layer, we exploit CNN features from hierarchical layers in order to make full use of the high-level structural information as well as preserving the spatial information of target.

### 2.2 Feature selection

In this paper, we employ CNN features extracted from VGG Net [28], which is trained on the large-scale ImageNet dataset; note that other CNN models may also be used alternatively, such as AlexNet [29] and R-CNN [30].

VGG-19 Net (with 16 convolutional layers and 3 full connect layers) has more deep structure than other CNN models, which can provide more semantic information. Given an input image frame, due to the CNN pooling propagation, the spatial resolution of each layer is more and more smaller, for instance, pool1 with the size of $224 \times 224$ and pool5 with only the size of only $7 \times 7$. The target in small size layers is hard to tell, so there is a need to resize each layer as a fixed size in order to locate the target accurately.

In this paper, we resize different layers to a constant size of $224 \times 224$ by using bilinear interpolation [31],

$$f_k = \sum_i \omega_{ki} F_i \tag{1}$$

where the weight $\omega_{ki}$ depends on the position of $k$ and $i$ neighboring feature vectors, and $F$ denotes the feature space.

As discussed above, in order to utilize CNN features from multi-layers, we choose conv2-2, conv3-4, and conv5-4 layers as feature representations specifically.

However, CNN features are pre-trained mainly aimed at dealing with classification tasks, so there are plenty of neurons used in describing generic object, which results in a very large number of wasted features. Here, by wasted features, we mean features which are redundant in discriminating target from background, especially when target deformation occurs. Furthermore, deeper CNN features are high-dimensional features (e.g., 512 dimension for conv5-4), leading to an extremely high computational complexity.

In order to alleviate the influences of these wasted features, it is of great importance to adopt an appropriate selection mechanism. From the experimental observation, we found that most redundant features have zero values for representing the target, so we adopt a sparse method to remove the redundancy similar to [25] and choose the feature with the largest coefficient as the template set.

### 2.3 Sparse representation model with CNN features and incremental subspace constraint

Motivated by the above dissussions, we propose a novel sparse model with CNN features (Fig. 1). Similar to [32], we assume that the target observation $\mathbf{z} \in \mathbb{R}^D$ can be sparsely representated by target template set $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N] \in \mathbb{R}^{D \times N}$ and the trivial template set $\mathbf{I} \in \mathbb{R}^{D \times D}$, where $D$ is the dimension of the observation vector, $N$ is the number of target templates, and $\mathbf{I}$ is an identity matrix; traditional sparse representation-based trackers approximate target object by linearly combining $\mathbf{M}$ and $\mathbf{I}$ with sparse constraints,

$$\operatorname{argmin}_{\mathbf{a}} \frac{1}{2} \| \boldsymbol{z} - \boldsymbol{A}\boldsymbol{a} \|_2^2 + \lambda \| \boldsymbol{a} \|_1, \ \ s.t. \mathbf{a}_M \geq 0, \tag{2}$$

where $\mathbf{A} = [\mathbf{M}, \mathbf{I}]$, $\mathbf{a} = [\mathbf{a}_M, \mathbf{a}_I] \in \mathbb{R}^{D+N}$ indicates the

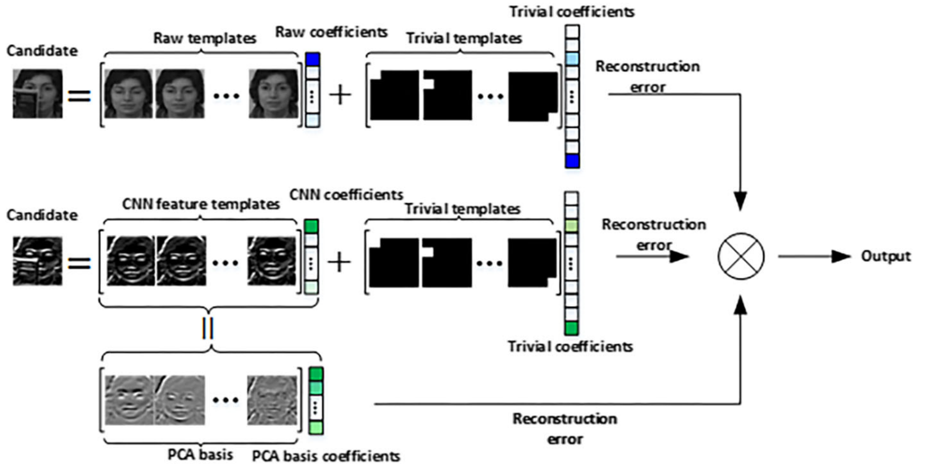

**Fig. 1** Overall sparse model with CNN features

corresponding sparse coefficients, and $\lambda$ controls the amount of regularization. The optimal state is the state with smallest reconstruction error.

However, traditional sparse representation-based trackers have some drawbacks. First, the computational complexity is relatively high which limit the real-time application. Secondly, they only use image intensity to construct the target template set, which can hardly handle the drastic appearance changes of target in practical visual tracking due to the lack of feature description. Thirdly, the target templates are only obtained from a previous couple of time instants, which cannot effectively obtain the underlying properties for modeling target appearance.

To solve the second problem, we use CNN features in describing the object. However, CNN features from different convolutional layers have different characteristics, high-level features have more distinguishing capabilities while low spatial resolution, and low-level features have more discriminative capabilities while sensitive to appearance changes. So, we construct a target template set by using hierarchical CNN features as a more complete feature descriptor. Furthermore, CNN features have a very high dimension in contrast to image intensity-based trackers, which results in an extreme computation complexity. In addition, most CNN features have barely contributed to effectively determine the exact location of the target, so we adopt a feature selection method to alleviate the redundancy.

To solve the third problem, an eigen template model is introduced for its ability to learn the temporal correlation of target appearances effectively from the past observation data by incremental update procedure, which compactly capture both rich and redundant image properties [33]. The incremental visual tracking (IVT) [34] algorithm can efficiently learn and update a low-dimensional PCA subspace representation of the target object and update the sample mean, which makes full use of the past observed target appearances. Experimental results have demonstrated that incremental learning of PCA subspace representation can deal with appearance changes caused by rotation, illumination variation, deformation and scale change efficiently. However, it has also been demonstrated that the performance of IVT tracker declines when partial occlusion occurs. Since the underlying assumption of PCA is that the error of each pixel is Gaussian distributed with small variances, this assumption does not hold anymore case of partial occlusion occurs. Furthermore, the IVT tracker may also fail when the target overlaps with a similar object.

In [19], each patch can be linearly represented by the eigenvectors corresponding to itself and the coefficients of almost all other eigenvectors will be zero; hence, by introducing $l_1$ regularization into the PCA reconstruction and modeling the error term $\mathbf{e}$ with arbitrary but sparse noise,

$$\underset{\mathbf{z},\mathbf{e}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{U}\boldsymbol{q}-\boldsymbol{e}\|_2^2+\tau\|\boldsymbol{e}\|_1 \tag{3}$$

where $\mathbf{U}\in\mathbb{R}^{D\times P}$ is the PCA eigen basis matrix and $P$ is the number of eigen basis vectors. $\mathbf{q}\in\mathbb{R}^{P}$ are the coefficients of $\mathbf{U}$ and $\tau$ controls the amount of regularization.

Motivated by [19], we model both reconstruction errors caused by sparse representation and the eigen subspace representation by solving

$$\begin{aligned}\underset{\mathbf{c}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2 \\ +\frac{\sigma}{2}\left\|(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})-\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}}(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})\right\|_2^2 \\ +\rho\|\boldsymbol{c}\|_1,\ s.t.\mathbf{c}_T\geq 0\end{aligned} \tag{4}$$

where $\mathbf{B} = [\mathbf{T}, \mathbf{I}]$, $\mathbf{T}$ is the target template set with CNN features, $\mathbf{c} = [\mathbf{c}_T, \mathbf{c}_I] \in \mathbb{R}^{3D+N}$ indicates the corresponding sparse coefficients, $\bar{\mathbf{t}}$ is the sample mean of target object, and $\sigma$ balances the contribution of the two terms.

This strategy constrains the reconstruction of the sparse representation to have a minimal reconstruction error in the PCA eigen basis representation. Meaning that our model with incremental subspace constrains can model both reconstruction errors caused by sparse representation and the eigen-subspace representation. This method constructs the reliable part of the target using a few number of PCA basis.

By integrating the subspace constrained sparse representation model with CNN features extracted and selected from hierarchical CNN layers, we get

$$\begin{aligned}\underset{\mathbf{a},\ \mathbf{c}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{Aa}\|_2^2+\lambda\|\boldsymbol{a}\|_1+\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2 \\ +\frac{\sigma}{2}\left\|(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})-\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}}(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})\right\|_2^2 \\ +\rho\|\boldsymbol{c}\|_1, s.t.\mathbf{a}_M, \mathbf{c}_T\geq 0\end{aligned} \tag{5}$$

The above overall model takes advantages of both the capability of hierarchical CNN features in describing the target and the subspace constrained sparse representation.

## 3 Optimization and the tracking algorithm

### 3.1 Optimization

Problem (5) can be decomposed into two sub-problems:

$$\underset{\mathbf{a}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{Aa}\|_2^2+\lambda\|\boldsymbol{a}\|_1, s.t.\mathbf{a}_M\geq 0 \tag{6 − 1}$$

$$\begin{aligned}\underset{\mathbf{c}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2 \\ +\frac{\sigma}{2}\left\|(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})-\boldsymbol{U}\boldsymbol{U}^{\boldsymbol{T}}(\boldsymbol{T}\boldsymbol{c}_{\boldsymbol{T}}-\bar{\boldsymbol{t}})\right\|_2^2 \\ +\rho\|\boldsymbol{c}\|_1, s.t.\mathbf{c}_T\geq 0\end{aligned} \tag{6 − 2}$$

Problem (6-1) can be solved by the LASSO method [35] and problem (6-2) can be solved by the accelerated proximal gradient (APG) method [16]. APG method is an effective approach to solve the following unconstrained minimization problem,

$$\min F(\boldsymbol{c}) + G(\boldsymbol{c}) \tag{7}$$

where $F(\mathbf{c})$ is a differentiable convex function with Lipschitz continuous gradient and $G(\mathbf{c})$ is a non-smooth convex function. We describe the details of solution as follows.

Let $\mathbf{R} = \mathbf{T} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{T}$ and $\mathbf{S} = \bar{\mathbf{t}}-\mathbf{U}\mathbf{U}^{\mathrm{T}}\bar{\mathbf{t}}$. Then the problem (6-2) can be reformed to the following formulation:

$$\begin{aligned}\underset{\mathbf{c}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2+\frac{\sigma}{2}\|\boldsymbol{S}-\boldsymbol{R}\boldsymbol{c}_T\|_2^2 \\ +\rho\|\boldsymbol{c}\|_1, s.t.\mathbf{c}_T\geq 0\end{aligned} \tag{8}$$

However, APG method cannot be used directly in our model since the original APG method is proposed for solving unconstrained minimization problem, so there is a need to covert our model into an unconstrained problem.

Let $1_T \in \mathbb{R}^N$ denotes the column vector with entries are all 1. Let $\psi$ ($\mathbf{c}$) denotes the indicator function defined by

$$\psi(\mathbf{c}) = \begin{cases} 0 & \boldsymbol{c}\geq 0 \\ +\infty & \text{otherwise} \end{cases}, \tag{9}$$

The problem (8) can be alternately reformed as the following unconstrained problem:

$$\begin{aligned}\underset{\mathbf{c}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2+\frac{\sigma}{2}\|\boldsymbol{S}-\boldsymbol{R}\boldsymbol{c}_T\|_2^2 \\ +\rho 1_T^T\boldsymbol{c}_T+\rho\|\boldsymbol{c}_{\boldsymbol{I}}\|_1+\psi(\mathbf{c}_{\boldsymbol{T}})\end{aligned} \tag{10}$$

Then, we can use the APG approach to solve this minimization problem with

$$\begin{aligned}F(\boldsymbol{c}) &= \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{Bc}\|_2^2+\frac{\sigma}{2}\|\boldsymbol{S}-\boldsymbol{R}\boldsymbol{c}_T\|_2^2+\rho 1_T^T\boldsymbol{c}_T, \\ G(\boldsymbol{c}) &= \rho\|\boldsymbol{c}_{\boldsymbol{I}}\|_1+\psi(\mathbf{c}_{\boldsymbol{T}}),\end{aligned} \tag{11}$$

In the above formulation, we need to solve an optimization problem:

$$\begin{aligned}c_{k+1} = \operatorname{argmin}_c\frac{L}{2}\left\|\mathbf{c}-\beta_{k+1}+\nabla F\left(\beta_{k+1}\right)/L\right\|_2^2 \\ +G(\mathbf{c}),\end{aligned} \tag{12}$$

where $k$ denotes the current iteration time, $L$ is the Lipschitz constant and $\beta_{k+1}$ is defined in Algorithm 1. We define $g_{k+1} = \beta_{k+1} - \nabla F(\beta_{k+1})/L$ and the soft-thresholding operator $\mathfrak{T}_\rho(x) = \operatorname{sign}(x)\max(|x|-\rho, 0)$. Then, the fast numerical algorithm for solving problem (6-2) is given in Algorithm 1.

**Algorithm 1:** Fast numerical algorithm

```
Set $c_0 = c_{-1} = \mathbf{0} \in \mathbb{R}^{3D+M}$ and set $\rho_0 = \rho_{-1} = 1$.
For $k$=0,1,…, until converge or a maximal number of iterations have been met
    $\beta_{k+1} = c_k + \frac{\rho_{k-1}-1}{\rho_k}(c_k - c_{k-1})$
    $g_{k+1}|_T = \beta_{k+1}|_T - \left(\left(\mathbf{B}^{\mathrm{T}}(\mathbf{B}\beta_{k+1} - \mathbf{y})\right)|_T - \mathbf{R}^{\mathrm{T}}(\mathbf{R}(\beta_{k+1}|_T) - \mathbf{S}) - \rho\mathbf{1}_T\right)/L$
    $g_{k+1}|_I = \beta_{k+1}|_I - \left(\mathbf{B}^{\mathrm{T}}(\mathbf{B}\beta_{k+1} - \mathbf{y})\right)|_I/L$
    $c_{k+1}|_T = \max(0, g_{k+1}|_T)$
    $c_{k+1}|_I = \mathfrak{T}_{\rho/L}(g_{k+1}|_I)$
    $\rho_{k+1} = (1 + \sqrt{1 + 4\rho_k^2})/2$
End
Obtain $\mathbf{c}$ via $\mathbf{c} = c_{k+1}$.
```

### 3.2 Particle filter tracking framework

Similar to [19], our method is based on Bayesian filtering framework in a Markov model. In a particle filter framework, given a set of observed image vectors $\mathbf{Z}_{1:t-1} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{t-1}]$, the posterior probability can be recursively computed as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|z_{t-1})d\mathbf{x}_{t-1}, \tag{13}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the dynamic model and $\mathbf{x}_t$ indicates the state vector.

At time $t$, by using Bayes rule, we get

$$p(\mathbf{x}_t|\mathbf{Z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|z_{t-1})d\mathbf{x}_{t-1}}{p(\mathbf{z}_t|z_{1:t-1})}, \tag{14}$$

where $p(\mathbf{z}_t|\mathbf{x}_t)$ denotes the observation likelihood of observing $\mathbf{z}_t$ at state $\mathbf{x}_t$. The state variable $\mathbf{x}_t$ is composed of six parameters $\mathbf{x}_t = [t_x, t_y, \theta_t, s_t, \delta_t, \phi_t]^{\mathrm{T}}$, where $t_x, t_y, \theta_t, s_t, \delta_t, \phi_t$ denote $x, y$ translations, rotation angle, scale, aspect ratio, and skew respectively.

The dynamic model is modeled by the Gaussian distribution,

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = N\left(\mathbf{x}_t; \mathbf{x}_{t-1}, \sum\right), \tag{15}$$

where $\Sigma$ is a diagonal covariance matrix.

Through the above method, we generate the candidates state set $\mathbf{X}_t = \left\{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^n\right\}$, where $n$ is the number of candidates sampled at each frame. For each particle $\mathbf{x}_t^i$, we crop out the related image region to get $\mathbf{z}_t^i$. Then, we get a candidate set $\mathbf{Z}_t = \left[\mathbf{z}_t^1, \mathbf{z}_t^2, \ldots, \mathbf{z}_t^n\right] \in R^{D\times n}$.

For each candidate, we solve the optimization problem using Algorithm 1. Then, the observation likelihood of state $\mathbf{x}_t^i$ is given as

$$\begin{aligned} p\left(\mathbf{z}_t^i, \boldsymbol{y}_t^i|\mathbf{x}_t^i\right) = \frac{1}{\varphi}&\exp\left(-\delta_1\left\|\mathbf{z}_t^i - \mathbf{M}\mathbf{a}_M^i\right\|_2^2\right) \\ &\times \exp\left(-\delta_2\left\|\mathbf{y}_t^i - \mathbf{T}\mathbf{c}_T^i\right\|_2^2\right) \\ &\times \exp\left(-\delta_3\left\|\mathbf{U}\mathbf{U}^{\mathrm{T}}\left(\mathbf{T}\mathbf{c}_T^i - \bar{\mathbf{t}}\right) - \left(\mathbf{T}\mathbf{c}_T^i - \bar{\mathbf{t}}\right)\right\|_2^2\right), \end{aligned} \tag{16}$$

where $\varphi$ is a normal factor, $\delta_1$, $\delta_2$, and $\delta_3$ balance the contributions between the three terms. The first term is the reconstruction error of the original image patch. The second term is the reconstruction error of sparse representation by target templates with CNN features. The third term reflects the relevancy between the reconstructed target and target PCA basis with CNN features. The optimal state $\mathbf{x}_t^*$ of frame $t$ is achieved by

$$\mathbf{x}_t^* = \underset{x_t^i \in \mathbf{x}_t}{\operatorname{argmax}}\, p\left(\boldsymbol{z}_t^i, \boldsymbol{y}_t^i \middle| \boldsymbol{x}_t^i\right) \tag{17}$$

### 3.3 Template update

To alleviate the influences of object appearance changes, there is a need to update the target template and PCA basis dictionary dynamically.

First, we use the method proposed in [27] to update target template. This updating strategy can effectively alleviate the influences caused by noise and occlusion. However, the target template is achieved from a previous couple of time, so they are not capable of dealing with numerous appearance variations due to the lack of long-term adjustment.

Then, we update the PCA basis dictionary using the method proposed in [34] and replace the oldest target template with the PCA reconstruction of the optimal candidate when the estimated optimal state is achieved. The PCA eigen template model could learn the temporal correlation of object appearances by incremental SVD update procedure effectively, so it has the ability to cover a long period of appearance changes relatively.

By adopting the strategy proposed above, we co-update the target template and PCA basis for current target appearance and long-term adjustment to improve the performance of our tracker.

## 4 Experiments

In order to illustrate the performance of our tracker, we test the robustness of our algorithm on 12 challenging video sequences with other 9 state-of-the-art trackers. The trackers are L1APG [16], ASLA [18], MTT [36], LSK [37], SST [38], IVT [34], FRAG [39], KMS [40], and SRUCK [41].

The proposed algorithm in this paper is implemented in MATLAB 2014a on a PC with Intel i7-4790 CPU (3.6 GHz) and 16 GB RAM memory. Before the experiment, we adopt some parameters and modify them according to other cited published works. For example, the iteration number is set to be 5 in the optimization part, but the tracking performance improves little when it is set to 10 or other lager value and the computational cost also increases with the iteration number. We did many experiments to choose the best parameter values as follows. Each sample is resized to $24 \times 24$ pixels. The number of target templates is set to be 11 with one fixed template extracted from the first frame. The candidate number $n$ in each frame is 600. These values mainly affect the speed of the tracker; we choose them to achieve a balance between the speed and the tracking performance. The number of PCA basis is set to be 10. The regularization factor $\lambda$ is set to be 0.01, σ is set to be 0.1, and ρ is set to be 0.01. The balance factors $\delta_1$, $\delta_2$, and $\delta_3$ are set to be 10, 10, and 1. The Lipschitz constant $L$ is set to be 8.

### 4.1 Qualitative evaluation

To evaluate our tracker with other state-of-the-art methods qualitatively, we choose 12 video sequences for testing. The 12 video sequences pose many challenging problems; Table 1 lists the characteristics of the sequences used in this paper.

Compared with traditional sparse representation-based trackers (e.g., L1APG, MTT, ASLA, LSK, and SST), our tracker outperforms in a wide range of challenging scenarios, especially when occlusion, rotation, and deformation occurs. This mainly attributes to our tracker that exploits both the advantage of sparse representation and incremental subspace learning, as well as using CNN features for representing the target. The incremental learning of PCA subspace representation method mainly aims at dealing with appearance changes caused by rotation, deformation, and scale variation, but it is sensitive to occlusion. In our algorithm, the occluded pixels of target object can be represented by the trivial templates, so when partial occlusion occurs, our tracker is more robust than the IVT tracker. Traditional sparse representation-based trackers are sensitive to rotation and deformation because they only use the image intensity for representing the target, but our method adopts CNN features from hierarchical layers in order to make full use of the high-level structural information as well as preserving the spatial information of target, so our tracker is more robust than these traditional L1 trackers.

**Table 1** Tracking sequences used in this paper

| Sequences | Frame | Main problem |
|---|---|---|
| Deer | 71 | Motion blur, fast motion, background clutters |
| David2 | 537 | Rotation |
| Crossing | 120 | Deformation, background clutters, scale variation |
| Boy | 602 | Scale variation, motion blur, rotation, fast motion |
| David3 | 252 | Rotation, background clutters, deformation, occlusion |
| FaceOcc1 | 892 | Occlusion |
| Football | 362 | Background clutters, Rotation, Occlusion |
| Walking | 412 | Low resolution, occlusion, scale variation, deformation |
| Football1 | 74 | Rotation, background clutters |
| Sylvester | 1345 | Illumination variation, rotation |
| Subway | 175 | Occlusion, background clutters, deformation |
| Mhyang | 1490 | Background clutters, deformation, illumination variation |

For example, in Fig. 2c–k, the target suffers from partial or total occlusion. In these scenarios, the IVT tracker presents bad performance, while our tracker can cope with these situations effectively. In Fig. 2a–i, the targets suffer from drastic appearance changes. Our tracker can still handle these situations effectively because the adopting of CNN features can utilize more structural information, while other L1 trackers failed in most of these situations.

In conclusion, our tracker performs well in all the 12 sequences while other 9 state-of-the-art trackers fail in some sequences.

### 4.2 Quantitative evaluation

We provide quantitative comparisons of our tracker with other state-of-the-art methods in terms of center location error (CLE) and overlap rate (VOR). The CLE is measured

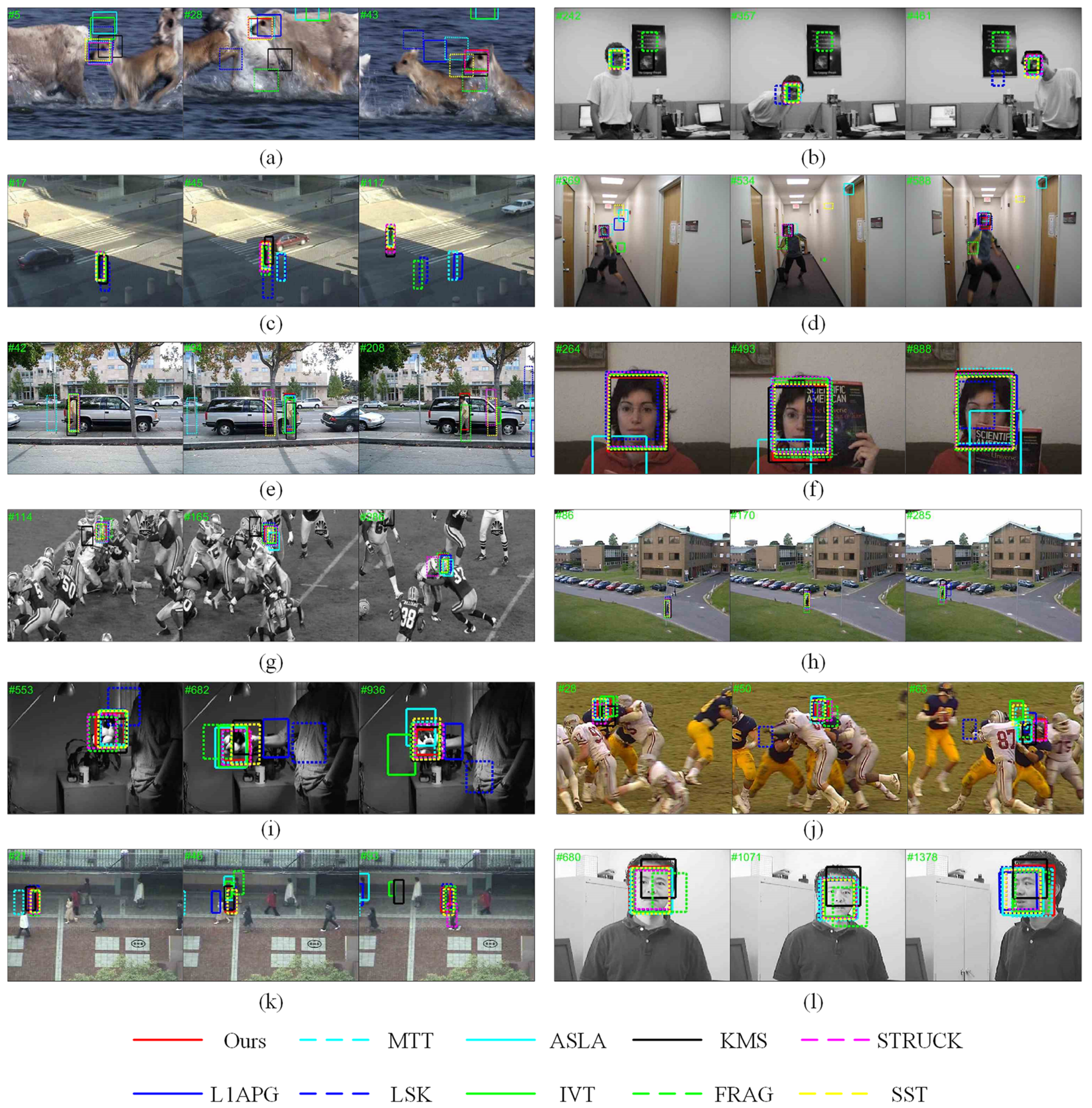

**Fig. 2 a–l** Qualitative comparisons. Tracking results of our algorithm and other 9 state-of-the-art tracking method on some representative frames of 12 sequences (deer, david2, crossing, boy, david3, faceOcc1, football, walking, sylvester, football1, subway, and mhyang, from left to right and top to bottom). Result of our method is marked with red rectangle

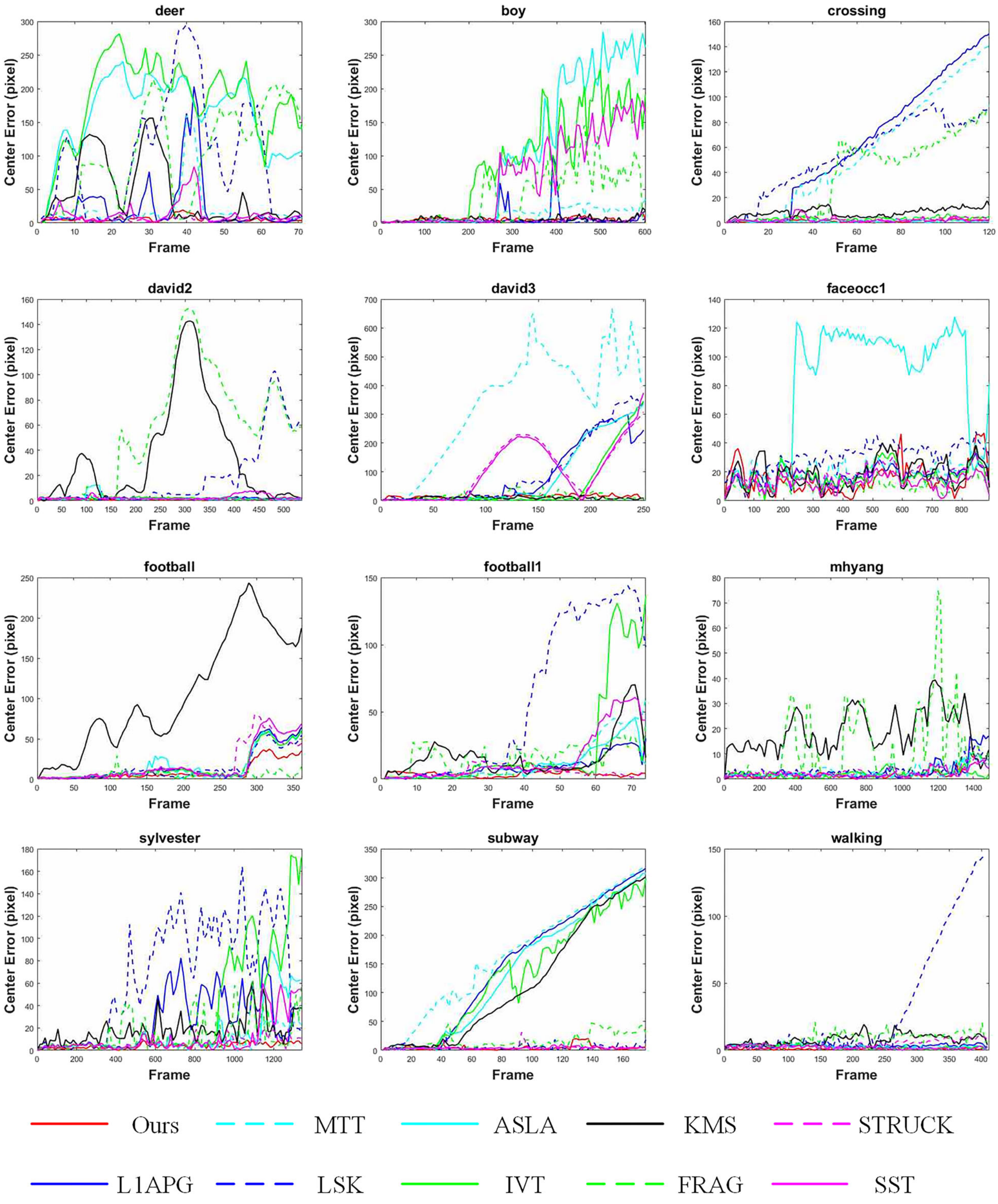

**Fig. 3** Center error comparisons on 12 sequences with 9 state-of-the-art trackers

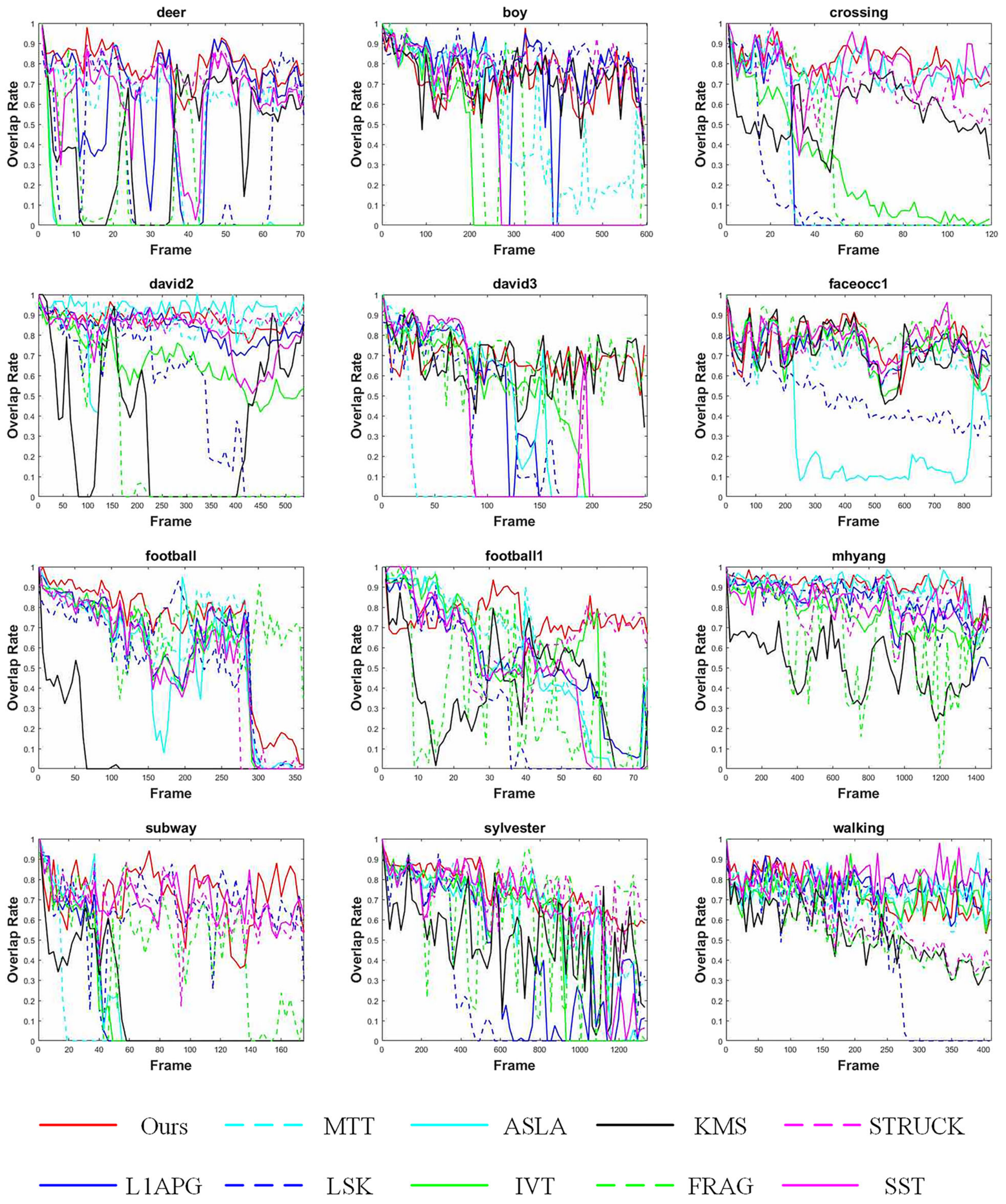

**Fig. 4** Overlap rate comparisons on 12 sequences with 9 state-of-the-art trackers

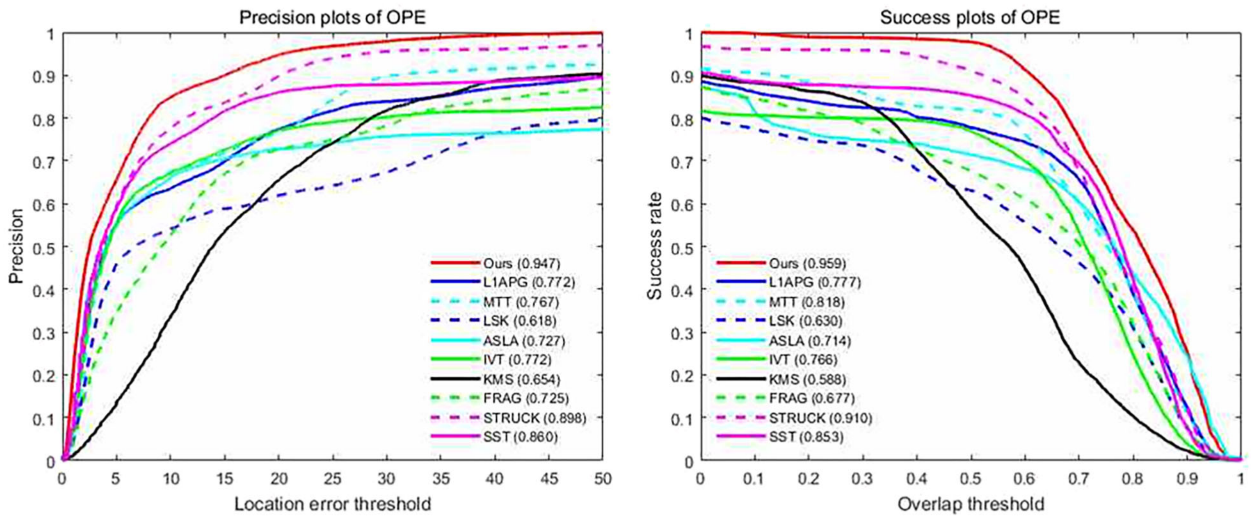

**Fig. 5** Precision and success plots on 12 sequences with 9 state-of-the-art trackers

by the Euclidean distance between the estimated target center location and the ground truth center location. The VOR is defined by $\frac{\text{area}(B_T \cap B_{GT})}{\text{area}(B_T \cup B_{GT})}$, where $B_T$ is the estimated target bounding box and $B_{GT}$ is the ground truth bounding box.

Figures 3 and 4 show the center error plot and the overlap rate plot of different trackers for each video sequence.

In addition, we adopt the precision and success rate for evaluating the tracking performance. The precision criteria is the percentage of frames which estimated location is within a given threshold distance of the ground truth and the success criteria is the ratios of successful frames at a given threshold ranged from 0 to 1.

Figure 5 shows the precision and success plots. The threshold of distance precision is 20 pixels and the threshold of overlap success rate is 0.5. Both precision plots and success plots show that our tracker is more robust than other state-of-the-art trackers over the 12 video sequences.

Tables 2 and 3 demonstrate the average center error and overlap rate of different tracking methods on each sequence. The best three results are marked in red, blue, and green fonts.

Note that in 6 of the 12 sequences (e.g., deer, david2, football1, mhyang, sylvester, and walking), the proposed tracker achieves the best average

**Table 2** Average center error for each sequence with 9 state-of-the-art trackers

| Sequences | L1APG | MTT | LSK | ASLA | IVT | KMS | FRAG | STRUCK | SST | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| deer | 24.19 | 18.92 | 98.77 | 160.11 | 182.85 | 43.88 | 105.09 | 5.27 | 13.84 | 5.12 |
| Boy | 7.03 | 12.77 | 2.24 | 106.26 | 92.42 | 4.83 | 40.52 | 3.84 | 66.97 | 3.64 |
| crossing | 63.43 | 57.15 | 54.82 | 1.38 | 3.44 | 8.40 | 38.59 | 2.81 | 2.32 | 1.40 |
| david2 | 1.38 | 1.71 | 18.87 | 1.80 | 1.91 | 34.95 | 56.87 | 1.50 | 2.73 | 0.91 |
| david3 | 90.00 | 341.34 | 104.87 | 87.81 | 52.18 | 9.61 | 13.55 | 106.50 | 104.51 | 13.43 |
| faceOcc1 | 17.33 | 21.00 | 30.41 | 78.29 | 17.43 | 19.16 | 10.97 | 18.78 | 13.01 | 15.80 |
| football | 15.11 | 13.67 | 14.01 | 15.46 | 14.84 | 105.55 | 5.36 | 17.31 | 17.21 | 8.45 |
| football1 | 9.20 | 12.74 | 58.22 | 11.17 | 24.14 | 17.05 | 15.70 | 5.46 | 15.73 | 3.70 |
| mhyang | 3.23 | 3.07 | 3.43 | 2.58 | 2.03 | 19.08 | 12.51 | 2.59 | 2.21 | 1.96 |
| sylvester | 26.24 | 7.55 | 68.43 | 15.75 | 34.43 | 18.08 | 15.00 | 6.30 | 11.18 | 3.95 |
| subway | 147.78 | 165.89 | 6.01 | 137.45 | 129.85 | 117.02 | 15.79 | 4.47 | 4.02 | 4.34 |
| walking | 3.31 | 3.47 | 33.87 | 2.57 | 2.23 | 8.32 | 9.31 | 4.63 | 2.12 | 1.39 |
| **Average** | 34.02 | 54.94 | 41.16 | 51.72 | 46.48 | 33.83 | 28.27 | 14.96 | 21.32 | 5.34 |

**Table 3** Average overlap rate for each sequence with 9 state-of-the-art trackers. The last row shows comparison results regarding the computational loads in terms of fps

| Sequences | L1APG | MTT | LSK | ASLA | IVT | KMS | FRAG | STRUCK | SST | Ours |
|---|---|---|---|---|---|---|---|---|---|---|
| deer | 0.62 | 0.62 | 0.27 | 0.03 | 0.03 | 0.42 | 0.17 | 0.75 | 0.63 | 0.80 |
| Boy | 0.75 | 0.50 | 0.82 | 0.37 | 0.25 | 0.72 | 0.39 | 0.76 | 0.37 | 0.78 |
| crossing | 0.21 | 0.20 | 0.13 | 0.77 | 0.28 | 0.55 | 0.31 | 0.69 | 0.77 | 0.82 |
| david2 | 0.84 | 0.86 | 0.50 | 0.90 | 0.66 | 0.35 | 0.24 | 0.87 | 0.80 | 0.88 |
| david3 | 0.38 | 0.10 | 0.36 | 0.42 | 0.47 | 0.67 | 0.68 | 0.29 | 0.30 | 0.69 |
| faceOcc1 | 0.75 | 0.70 | 0.48 | 0.32 | 0.73 | 0.73 | 0.82 | 0.73 | 0.78 | 0.75 |
| football | 0.57 | 0.60 | 0.55 | 0.54 | 0.56 | 0.08 | 0.71 | 0.55 | 0.53 | 0.69 |
| football1 | 0.53 | 0.55 | 0.32 | 0.51 | 0.55 | 0.39 | 0.37 | 0.67 | 0.49 | 0.75 |
| mhyang | 0.82 | 0.85 | 0.83 | 0.90 | 0.78 | 0.53 | 0.65 | 0.82 | 0.82 | 0.91 |
| sylvester | 0.41 | 0.65 | 0.24 | 0.60 | 0.52 | 0.48 | 0.59 | 0.73 | 0.63 | 0.74 |
| subway | 0.16 | 0.07 | 0.67 | 0.19 | 0.16 | 0.16 | 0.47 | 0.66 | 0.69 | 0.71 |
| walking | 0.77 | 0.72 | 0.46 | 0.75 | 0.73 | 0.53 | 0.55 | 0.59 | 0.81 | 0.78 |
| **Average** | 0.57 | 0.53 | 0.47 | 0.53 | 0.48 | 0.47 | 0.50 | 0.68 | 0.64 | 0.77 |
| **FPS** | 2.1 | 1.2 | 5.3 | 8.8 | 31.1 | 186.8 | 4.13 | 22.4 | 1.3 | 3.0 |

center error. In 7 of the 12 sequences (e.g., deer, crossing, david3, football1, mhyang, sylvester, and walking), the proposed tracker achieves the best overlap rate. In other sequences, our tracker achieves either the second or the third best scores. The proposed tracker also achieves both the best scores in average center error and average overlap rate for all the 12 sequences, meaning that our tracker outperforms other state-of-the-art trackers in many challenging situations significantly.

## 5 Conclusions

In this paper, we propose a robust L1 tracker with CNN features. Different from traditional sparse representation-based tracking algorithms, our model not only exploits convolutional features to improve the robustness for describing the object appearance but also uses the trivial templates to model both reconstruction errors caused by sparse representation and the eigen-subspace representation. A customized APG method is developed to solve the optimization problem effectively. Both qualitative and quantitative evaluations demonstrate that our tracker outperforms other state-of-the-art trackers in many challenging situations.

**Acknowledgements**
The authors would like to thank Image Engineering & Video Technology Lab for the support.

**Authors' contributions**
HW and TX came up with the algorithm and improved the algorithm. HW wrote and revised the paper. Both authors read and approved the final manuscript.

**Funding**
This work was supported by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802, the General Program of National Nature Science Foundation of China under Grant 61371132 and 61471043, and the International S&T Cooperation Program of China under Grant 2014DFR10960.

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

## Publisher's Note

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

Received: 27 September 2017 Accepted: 6 November 2017
Published online: 21 November 2017

**References**

1. L Baroffio, A Canclini, MCA Redondi, M Tagliasacchi, G Dán, E Eriksson, V Fodor, J Ascenso, P Monteiro, *Enabling Visual Analysis in Wireless Sensor Networks, IEEE International Conference on Image Processing* (2015), pp. 3408–3410
2. F Zhao, B Li, H Chen, X Lv, Joint beamforming and power allocation for cognitive MIMO systems under imperfect CSI based on game theory. Wirel. Pers. Commun. **73**, 679–694 (2013)
3. F Zhao, X Sun, H Chen, R Bie, Outage performance of relay-assisted primary and secondary transmissions in cognitive relay networks. EURASIP J. Wirel. Commun. Netw. **2014**, 60 (2014)
4. F Zhao, L Wei, H Chen, Optimal time allocation for wireless information and power transfer in wireless powered communication systems. IEEE Trans. Veh. Technol. **65**, 1830–1835 (2016)
5. A Yilmaz, O Javed, M Shah, Object tracking: a survey. ACM Comput. Surv. **38**, 13 (2006)

6. Y Wu, J Lim, MH Yang, *Online Object Tracking: A Benchmark, IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 2411–2418
7. H Yang, L Shao, F Zheng, L Wang, Z Song, Recent advances and trends in visual tracking: a review. Neurocomputing **74**, 3823–3831 (2011)
8. AW Smeulders, DM Chu, R Cucchiara, S Calderara, A Dehghan, M Shah, Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1442–1468 (2013)
9. X Li, W Hu, C Shen, Z Zhang, A Dick, AVD Hengel, A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. **4**, 58 (2013)
10. Y Yin, X Wang, D Xu, F Liu, Y Wang, W Wu, Robust visual detection–learning–tracking framework for autonomous aerial refueling of UAVs. IEEE Trans. Instrum. Meas. **65**, 510–521 (2016)
11. W Zhong, H Lu, MH Yang, Robust object tracking via sparse collaborative appearance model. IEEE Trans. Image Process. **23**, 2356–2368 (2014)
12. Y Yin, D Xu, X Wang, M Bai, Online state-based structured SVM combined with incremental PCA for robust visual tracking. IEEE Trans. Cybern. **45**, 1988–2000 (2017)
13. S Chen, S Li, R Ji, Y Yan, S Zhu, Discriminative local collaborative representation for online object tracking. Knowl.-Based Syst. **100**, 13–24 (2016)
14. S Zhang, H Yao, X Sun, X Lu, Sparse coding based visual tracking: review and experimental comparison. Pattern Recogn. **46**, 1772–1788 (2013)
15. X Mei, H Ling, Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 2259 (2011)
16. H Ji, H Ling, Y Wu, C Bao, *Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach, IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 1830–1837
17. T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, IEEE Conference on Computer Vision and Pattern Recognition. **157**, 2042-2049 (2012).
18. X Jia, *Visual Tracking Via Adaptive Structural Local Sparse Appearance Model, IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 1822–1829
19. D Wang, H Lu, MH Yang, Online object tracking with sparse prototypes. IEEE Trans. Image Process. **22**, 314 (2013)
20. B Liu, J Huang, C Kulikowski, L Yang, Robust visual tracking using local sparse appearance model and K-selection. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 2968–2981 (2013)
21. J Guo, T Xu, G Shi, Z Rao, X Li, Multi-view structural local subspace tracking. Sensors **17**, 666 (2017)
22. H Wang, S Zhang, Y Du, H Ge, B Hu, Visual tracking via probabilistic collaborative representation. J. Electron. Imaging **26**, 013010 (2017)
23. H Kim, S Jeon, S Lee, JK Paik, *Robust Visual Tracking Using Structure-Preserving Sparse Learning, IEEE Signal Processing Letters, PP* (2017), pp. 1–1
24. C Ma, JB Huang, X Yang, MH Yang, *Hierarchical Convolutional Features for Visual Tracking, IEEE International Conference on Computer Vision* (2016), pp. 3074–3082
25. L Wang, W Ouyang, X Wang, H Lu, *Visual Tracking with Fully Convolutional Networks, IEEE International Conference on Computer Vision* (2016), pp. 3119–3127
26. M Danelljan, G Häger, FS Khan, M Felsberg, *Convolutional Features for Correlation Filter Based Visual Tracking, IEEE International Conference on Computer Vision Workshop* (2015), pp. 621–629
27. Y Lecun, Y Bengio, G Hinton, Deep learning. Nature **521**, 436–444 (2015)
28. K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Computer Science, (2014).
29. A Krizhevsky, I Sutskever, GE Hinton, *ImageNet Classification with Deep Convolutional Neural Networks, International Conference on Neural Information Processing Systems* (2012), pp. 1097–1105
30. R Girshick, J Donahue, T Darrell, J Malik, *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, Computer Vision and Pattern Recognition* (2014), pp. 580–587
31. B Hariharan, P Arbeláez, R Girshick, J Malik, *Hypercolumns for Object Segmentation and Fine-Grained Localization* (2014), pp. 447–456
32. X Mei, H Ling, *Robust Visual Tracking Using L (1) Minimization, IEEE International Conference on Computer Vision* (2009), pp. 1436–1443
33. T Bai, YF Li, Robust visual tracking with structured sparse representation appearance model. Pattern Recogn. **45**, 2390–2404 (2012)
34. DA Ross, J Lim, RS Lin, MH Yang, Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**, 125–141 (2008)
35. R Tibshirani, Regression shrinkage and selection via the lasso: a retrospective. J. R. Stat. Soc. **73**, 267–288 (2011)
36. T Zhang, B Ghanem, S Liu, N Ahuja, Robust visual tracking via structured multi-task sparse learning. Int. J. Comput. Vis. **101**, 367–383 (2013)
37. C Kulikowsk, *Robust Tracking Using Local Sparse Appearance Model and K-Selection, IEEE Conference on Computer Vision and Pattern Recognition* (2011), pp. 1313–1320
38. T Zhang, S Liu, C Xu, Y Shuicheng, B Ghanem, N Ahuja, MH Yang, *Structural Sparse Tracking, IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 150–158
39. A Adam, E Rivlin, I Shimshoni, *Robust Fragments-Based Tracking Using the Integral Histogram, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference* (2006), pp. 798–805
40. D Comaniciu, V Ramesh, P Meer, Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 564–575 (2003)
41. S Hare, A Saffari, PHS Torr, *Struck: Structured Output Tracking with Kernels, IEEE International Conference on Computer Vision* (2011), pp. 263–270