Zhang et al. EURASIP Journal on Image and Video Processing (2019) 2019:20
https://doi.org/10.1186/s13640-019-0411-1

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Adaptive visual target tracking algorithm based on classified-patch kernel particle filter

Guangnan Zhang[1,2,3], Jinlong Yang[3*], Weixing Wang[1], Yu Hen Hu[4] and Jianjun Liu[3]

Abstract

We propose a high-performance visual target tracking (VTT) algorithm based on classified-patch kernel particle filter (CKPF). Novel features of this VTT algorithm include sparse representations of the target template using the label-consistent K-singular value decomposition (LC-KSVD) algorithm; Gaussian kernel density particle filter to facilitate candidate template generation and likelihood matching score evaluation; and an occlusion detection method using sparse coefficient histogram (ASCH). Experimental results validate superior performance of the proposed tracking algorithm over state-of-the-art visual target tracking algorithms in scenarios that include occlusion, background clutter, illumination change, target rotation, and scale changes.

Keywords: Visual target tracking, K-singular value decomposition, Sparse coding, Dictionary learning, Particle filter

1 Introduction

Visual target tracking (VTT) [1–15] is a key enabling technology for numerous emerging computer vision applications including video surveillance, navigation, human-computer interactions, augmented reality, higher level scene understanding, and action recognition among many others. It is a challenging task because the visual observations often suffer from interference due to occlusion, scale and shape variation, illumination variation, background clutter, and related factors.

VTT differs from conventional tracking task in that the observation at each time instant is a video frame and the motion trajectory is confined to the spatial coordinates in each frame. On the other hand, like the conventional tracking, a VTT algorithm is divided into the prediction phase and an update phase. In the prediction phase, a motion model is incorporated to predict the target location based on current estimate. In the updated phase, a maximum likelihood (ML) estimate of the target location is sought based on observations made in the current frame. Then, an updated target position is decided based on predicted location and the ML estimated position. These location predictions and estimations are traditionally realized using sequential Bayesian estimation algorithms such as Kalman filters or particle filters.

Based on how the ML estimation of target location is realized, current VTT algorithms may be categorized into two families: discriminative algorithms versus generative algorithms [5]. Discriminative methods detect the presence of a tracked object using a pattern classification approach with the objective to distinguish the foreground target from the background. For example, the multiple instance learning (MIL) [6] method puts all ambiguous positive and negative samples into bags to learn a discriminative model for visual target tracking. Generative methods detect the tracked object by searching for the region most resembling to the target model, based on templates or subspace models. In [7], a robust fragments-based tracking method is proposed to handle partial occlusions or pose changes. Every patch votes on the possible positions and scales of the target in the current frame by comparing the intensity histogram against the corresponding histogram of each image patch. However, a static appearance model of the target cannot adapt to rapid appearance changes of the target. Incremental learning visual tracking (IVT) algorithm [8] handles the problem of changing target appearance. In the template update process, a forgetting factor is

* Correspondence: yjlgedeng@163.com
[3]School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
Full list of author information is available at the end of the article

introduced to ensure that less modeling power is wasted fitting older observations. Visual tracking decomposition (VTD) algorithm [9] is proposed to handle the appearance and motion changes of the target occur at the same time. In the tracking process, the observation model is decomposed into multiple basic observation models that can cover different specific target appearances. The motion model is also represented by combining multiple basic models that cover different motion types. Then two types of basic models are used to construct the multiple basic trackers to handle a certain change of a target.

Tracking algorithms based on the sparse model have attracted great interests lately. Mei et al. [10, 11] formulated visual target tracking as a sparse approximation problem in the particle filtering (PF) framework [12, 13]. Using a dictionary of image patches, the target template can be represented as a weighted linear combination of very few (hence sparse representation) image templates in the dictionary. The sparse representation can be estimated by solving an l_1-norm regularized least squares (LS) problem. In [14], a real-time robust l_1 tracker is proposed by adding an l_2-norm regularization to the coefficients associated with the trivial templates, and an accelerated proximal gradient (APG) method is employed to speed up the problem solving. Multi-task tracking (MTT) is proposed [15] as a multi-task sparse learning problem in a PF framework. The particles are modeled as linear combinations of dictionary image templates, and the interdependencies between particles are exploited to improve the tracking performance. In [5], an adaptive structural local sparse appearance model is proposed to locate the target more accurately by considering the spatial information of the target based on an alignment-pooling method. Moreover, the incremental subspace learning and sparse representation are combined to update the template, which can adapt to the appearance change of the target with less possibility of drifting. When the target exhibits dramatic appearance changes, a collaborative model is proposed [16] that combines a sparsity-based discriminative classifier and a sparsity-based generative model. With this appearance model, both holistic updates and local representations are considered. Moreover, the latest observations and the original template are used to update the model and adapt to the appearance change while mitigating the drift problem.

Most of the dictionaries based on the sparse representation theory are constructed directly by the samples of the template base or obtained by the clustering method with some constraints. The image templates in the dictionary often lack the ability of discrimination. Moreover, the templates updated by the same update scheme cannot adapt to the changes of the foreground and the background of the target. To address these concerns, in this work, we propose an adaptive visual target tracking algorithm based on classified-patch kernel particle filter (CKPF), which has the following advantages:

(a) Classified patches and low-dimensional dictionary are considered in the CKPF. Note that low-dimensional dictionary and classification parameters (CP) are learned by the label-consistent K-SVD (LC-KSVD) [17, 18] technique. To the best of our knowledge, this is the first work to extend the LC-KSVD approach to exploit the intrinsic structure among the patches of the visual target. The image patches in the dictionary trained using LC-KSVD will be more discriminative to classify foreground from the background, and the obtained low dictionary can reduce the computational burdens.
(b) The anti-occlusion sparse coefficient histograms (ASCHs) [16] are merged in CKPF to enhance the ability of anti-occlusion. If the reconstructed error of one patch is bigger than the threshold, the patch will be marked as occluded, and the corresponding sparse coefficients were displaced with zero to reduce the negative influence.
(c) Gaussian kernel density (GKD) of the learned patches is considered to make the proposed algorithm more stable. The reason is that the importance of each patch is considered in the structure of candidate template according to the distance close to the center of the template.
(d) An adaptive template update scheme is developed to adapt to the target appearance changes improving the robustness of the tracker. It is because the appearance of the target often changes significantly due to the disturbance of illumination changes, occlusion, rotation, and scale variation. When the target is occluded, the arrived template usually cannot describe the real target effectively. Therefore, the weight of the arrived template should decrease at this time. Otherwise, the weight should increase due to the accurately estimate of the arrived template without other disturbance factors.

Our proposed visual target tracker differs from existing approaches [10–16] in several aspects, such as the dictionary learning of the local image patches by LC-KSVD, likelihood model construction of the candidate particles, as well as the design of the adaptive parameter for the template update. The main contributions of this paper are threefold. (a) Classification parameters and low-dimensional patches are learned by LC-KSVD to construct the CKPF. (b) Isotropic Gaussian kernel density of the patches is proposed to produce the mixture likelihood of the each candidate particle. (c) An adaptive

template update scheme is proposed to adapt to the target appearance changes.

The remainders of this paper are organized as follows. In Section 2, we summarize the details of the proposed adaptive visual target tracking algorithm based on CKPF. An overview of the LC-KSVD is presented. Meanwhile, adaptive template update scheme is developed and discussed. In Section 3, extensive simulation results comparing our proposed algorithm against existing visual target trackers are reported and the implications of these results are discussed. Conclusions and future works are presented in Section 4.

2 Methods

2.1 Overview of the algorithm

As shown in Fig. 1, the target is represented as a rectangular template in each frame, and the target template will be scaled to 32×32 pixels (big red boxes on the right). Candidate target region will also be scaled at the same ratio before further processing. Pixels within the template are assumed to be positive samples of the target. A 4-pixel wide strip surrounding the template is defined as the background whose edges outside and inside of the target template are denoted by B1 and B2, i.e., the gray annular area (B2-B1) with width 8 pixels are the background area. A patch is defined as a 6×6 square. $N_p = 196$ image patches (positive samples) will be extracted from the template (foreground, target), and $N_n = 196$ patches will be extracted from the background as negative sample. These extracted image patches are regularly distributed over the foreground or the background regions respectively with overlaps as needed. Together, the positive-labeled patches represent the target and the negative-labeled patches represent the background.

Each patch is raster-scanned and converted into a 36×1 vector. Hence, there are 196 vectors labeled with +1 (positive samples) and 192 vectors labeled with 0 (negative samples). We denote the total number of patches $N = 392$. A label-consistent, kernel singular value decomposition (LC-KSVD) algorithm will be applied to both the 196 positive vectors and the 196 negative vectors and select a subset of 50 vectors from each of them to form a labeled dictionary. This dictionary consists of 50 vectors with positive (+1) labels and 50 vectors with negative (0) labels. Let $K = 100$, the dictionary may be represented by a $36 \times K$ matrix $\mathbf{D}$. The dictionary will be estimated from the initial frame where the target to be tracked is specified for the tracking algorithm. It will remain unchanged until template update operation is performed.

The LC-KSVD algorithm also yields a sparse representation of each patch (36×1 vector) as a weighted combination of the 100 vectors selected in the dictionary. Two constraints are imposed on the potential sparse representations: (a) (discriminative constraint) Sparse vectors corresponding to foreground (or background) patches should have similar representation. This is represented by a discriminative parameter matrix $\mathbf{A}_{K \times K}$. (b) (classification constraint) Class labels (+1, 0) can be reproduced from weighted linear combination of the sparse representation. This is represented by a $2 \times K$ classification parameter matrix $\mathbf{W}$. In addition to the sparse representation of each foreground and background patches, represented by a $K \times N$ matrix $\mathbf{X}$, the LC-KSVD algorithm can estimate the dictionary $\mathbf{D}$, the discriminative parameter matrix $\mathbf{A}$ and the classification parameter matrix $\mathbf{W}$ simultaneously.

Given the dictionary $\mathbf{D}$ and sparse representation of the template $\mathbf{X}$, tracking begins by moving into the next

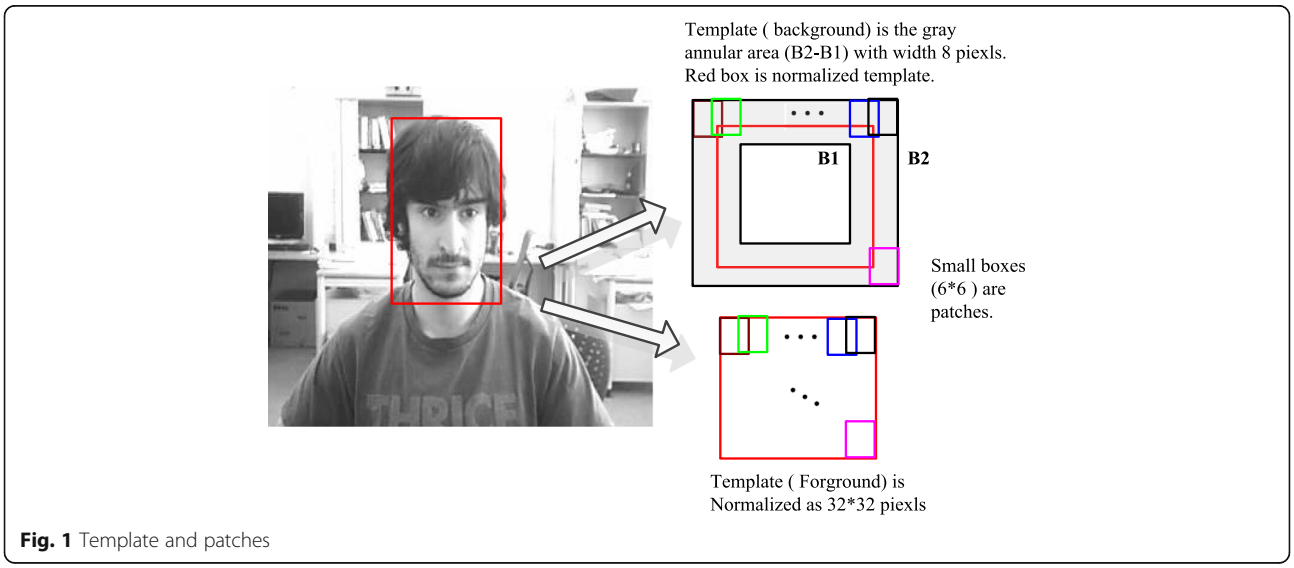

Fig. 1 Template and patches

frame. A kernel particle filter is applied to generate 100 potential target positions at $(k+1)$th frame according to the particle representation of the state transition probability $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ such that $E(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathbf{x}_k$ where $\mathbf{x}_k = \{x_k, y_k, \theta_k, s_k, \alpha_k, \beta_k\}$ is the state vector of the target at the kth frame. The assumption is that the target motion may be described by an affine transformation, for example, (x_k, y_k) is the target position, θ_k, s_k, α_k, and β_k are the rotation angle, the scaling factor, the aspect ratio, and the angle of inclination, respectively. We also assume $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ has a Gaussian distribution where the covariance matrix is selected based on prior knowledge of the tracking task.

Each particle corresponds to a candidate target template. Then, 196 image patches are extracted, and corresponding sparse representation **X'** are evaluated using LC-KSVD and the library **D**. A kernel density weighted sparse coefficient similarity score (SCSS) then will be applied to produce an estimate of the likelihood probability between the sparse representation of the template **X** and the current template candidate **X'**. The kernel density weightings place more weight on image patches that are closer to the center of the template and less weight on image patches on peripherals of the template. The location of the best-matched template will be designated as new target position.

Before moving into the next frame, the tracking algorithm may also adaptively update the template when occlusion of the target is detected. This is accomplished by using a sparse coefficient histogram matrix (SCHM) [16] to estimate the level of occlusion of the target. If so, the algorithm uses the newly estimated template, or a weighted linear combination of the estimated template and an initial template depending on the percentage of patches that are deemed occluded. With the newly updated template, the algorithm moves to the following frame.

A block diagram summarizing above overview of the proposed algorithm is depicted in Fig. 2. It has an initialization phase where a low-dimensional label-consistent dictionary **D** of image patches will be estimated, and the sparse representation **X** as well as classification parameters **W** of individual patches are also computed. Next, the kernel density-based particle filter (KPF) algorithm generates candidate templates in the following frame. For each candidate template, the likelihood score will be evaluated, and the maximum likelihood estimate of the target position will be computed. This is followed by an adaptive template update phase where occlusion of the target is detected.

2.2 Theoretical backgrounds

2.2.1 LC-KSVD

The LC-KSVD dictionary learning algorithm [17, 18] in Fig. 2 can simultaneously train an over-complete low-dimensional dictionary and a linear classifier, i.e., the obtained dictionaries have both reconstructive and discriminative abilities. The objective function is expressed as

$$\begin{aligned}\langle D, W, A, X\rangle = \underset{D,W,A,X}{\arg\min} \|Y - DX\|_2^2 + \alpha\|Q - AX\|_2^2 \\ + \beta\|H - WX\|_2^2, \quad \text{s.t. } \forall i, \|x_i\|_0 \le T_0\end{aligned} \tag{1}$$

where $Y = \{y_i\}_{i=1}^N \in \mathrm{R}^{n\times N}$ denotes the input sample set, $X = [x_1, x_2, \cdots, x_N] \in R^{K\times N}$ denotes the coefficient matrix, $D = [d_1, d_2, \cdots, d_K] \in \mathrm{R}^{n\times K}$ denotes the low-dimensional dictionary matrix containing $K \ll N$ prototype sample-atoms for columns $\{d_j\}_{j=1}^K$, and T_0 denotes the degree of sparsity. $Q \in R^{K\times N}$ denotes the sparse codes with discriminative power of Y for classification. A is a linear transformation matrix, which can transform the original sparse codes to be most discriminative in

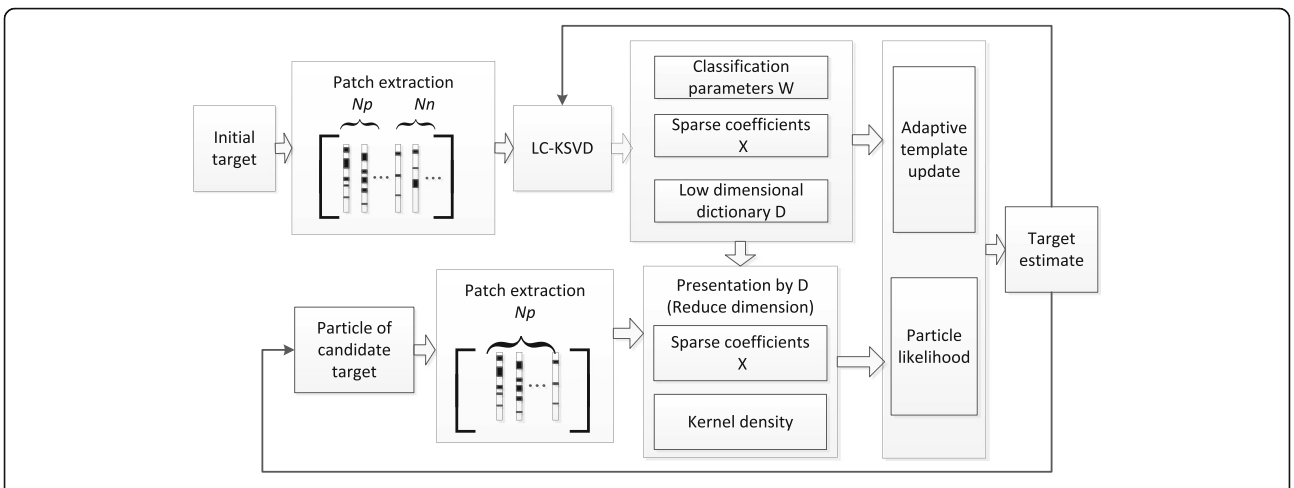

Fig. 2 Block diagram of the proposed algorithm

sparse feature space. $\|Q-AX\|_2^2$ denotes the discriminative sparse code error, which forces the samples with same class label to have the similar sparse representations. $\|H-WX\|_2^2$ denotes the classification error, W is the classification parameter matrix, and H is the class label of input samples. α and β are the scalars controlling the relative contribution of the corresponding terms [18].

The K-SVD method [19] can be used to obtain the optimal solutions for all the parameters simultaneously. Specifically, Eq. (1) can be rewritten as

$$\langle D, W, A, X\rangle = \arg\min_{D,W,A,X} \left\| \begin{pmatrix} Y \\ \sqrt{\alpha}Q \\ \sqrt{\beta}H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\alpha}A \\ \sqrt{\beta}W \end{pmatrix} X \right\|_2^2, \quad \text{s.t. } \forall i, \|x_i\|_0 \le T \tag{2}$$

Let $Y_{\text{new}} = (Y^{\mathrm{T}}, \sqrt{\alpha}Q^{\mathrm{T}}, \sqrt{\beta}H^{\mathrm{T}})^{\mathrm{T}}$, $D_{\text{new}} = (D^{\mathrm{T}}, \sqrt{\alpha}A^{\mathrm{T}}, \sqrt{\beta}W^{\mathrm{T}})^{\mathrm{T}}$, then Eq. (2) can be expressed as

$$\langle D_{\text{new}}, X\rangle = \arg\min_{D_{\text{new}},X} \{\|Y_{\text{new}} - D_{\text{new}}X\|_2^2\}, \quad \text{s.t. } \forall i, \|x_i\|_0 \le T \tag{3}$$

Then D_{new} can be obtained by using the K-SVD method, i.e., D, A, and W are learned simultaneously. More descriptions about LC-KSVD can refer to [17, 18].

In Eq. (1), the learned dictionary can be better used to represent the target due to the constraint terms. The discriminative sparse code error can force the samples with same class to have the similar sparse representations, which can enlarge the difference between classes of training data. Moreover, the classification error can effectively train a classifier to identify the foreground and background of the target.

2.2.2 Sparse coefficient histogram and occlusion detection

The patches of the target can be represented by using the obtained low dimensional dictionary D and the sparse coefficient of each patch can be used to construct the histogram matrix. However, some patches in the candidate target may be occluded, and the coefficient histogram cannot express the feature of candidate target accurately. As a result, the target cannot be estimated accurately. Taking this problem into account, the occlusion detection strategy [16] is employed according to the reconstruction error of each patch. And then the sparse coefficient histogram can be updated according to the occlusion detection results.

Assume that ξ_i denotes the sparse coefficient vector of the ith patch, we have

$$\min_{\xi_i} \|y_i - D\xi_i\|_2^2 + \lambda\|\xi_i\|_1 \tag{4}$$

The sparse coefficient histogram matrix can be established by concatenating the sparse coefficient vector ξ_i, i.e.,$\rho = [\xi_1, \xi_2, \ldots, \xi_{N_p}]$. If the target is partially occluded, then some of the patches of the target are occluded, and their corresponding sparse coefficients will be meaningless, which makes the sparse coefficient matrix ρ unable to express the candidate target well, causing big reconstruction error. Therefore, we introduce an occluded target detective mechanism to identify the occluded patches and their corresponding sparse coefficients.

It is defined that if the reconstructed error of each patch is bigger than the threshold, the patch will be marked as occluded, and then the corresponding sparse coefficient vector is reset to zero. The candidate histogram matrix after occlusion detection is defined as $\phi = \rho \odot o$, where $\odot$ denotes the element-wise multiplication. $o \in R^{(K_p+K_n)\times N_p}$ denotes the matrix of occluded detection, and o_i is the element of the matrix o, and can be defined as:

$$o_i = \begin{cases} 1, & \varepsilon_i < \varepsilon_0 \\ 0, \text{otherwise} \end{cases} \tag{5}$$

where $\varepsilon_i = \|y_i - D_t\xi_{i\_t}\|_2^2$ denotes the reconstructed error of the ith patch. Note that only the positive patches are used to compute the reconstructed error, therefore D_t denotes the dictionary which only consists of the set of positive patches from the learned dictionary D, ξ_{i_t} denotes the corresponding sparse coefficient vector of D_t, and ε_0 denotes the threshold of reconstructed error of each patch. If $\varepsilon_i \ge \varepsilon_0$, then the ith patch be considered as occluded and the corresponding coefficient vector is set as zero.

2.3 Classified-patch kernel particle filter

Given the observation set of target $y_{1:k} = \{y_1, y_2, \ldots, y_k\}$ up to the kth frame, the target state x_k can be extracted via the maximum posterior estimation, i.e., $\hat{x}_k = \arg\max_{x_k^i} p(x_k^i|y_{1:k})$, where x_k^i denotes the state of the ith sampled particle of the kth frame. The posterior probability $p(x_k^i|y_{1:k})$ can be inferred by the Bayesian recursion, i.e.,

$$p(x_k^i|y_{1:k}) \propto p(y_k|x_k) \int p(x_k|x_{k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1} \tag{6}$$

where $p(y_k|x_k)$ denotes the observation model. $p(x_k|x_{k-1})$ denotes the dynamic model which describes the temporal correlation of the target states between consecutive frames. The affine transformation with six parameters is

utilized to model the target motion between two consecutive frames. The state transition is formulated as $p(x_k|x_{k-1}) = N(x_k; x_{k-1}, \Sigma)$, where Σ is a diagonal covariance matrix whose elements are the variances of the affine parameters.

The observation model $p(y_k|x_k)$ denotes the likelihood of the observation y_k at state x_k. It plays an important role in robust tracking. In this paper, we aim to construct a robust likelihood model having the anti-occlusion ability and foreground target identification ability by merging the similarity of sparse coefficient histograms [16] and the classification information. Moreover, we consider the spatial information of each patch by using the isotropic Gaussian kernel density, which can keep the stability of the proposed algorithm for visual target tracking.

The likelihood of the lth particle is expressed as

$$p_l = \sum_{i=1}^{N_p} k\left(\left\|\frac{y_k^l - c_i}{h}\right\|^2\right) M_{k,i}^l L_{k,i}^l \tag{7}$$

where $M_{k,i}^l$ and $L_{k,i}^l$ denote the likelihood of classification and the similarity function of the target histograms between the candidate and the template. $k(\|\frac{y_k^l - c_i}{h}\|^2)$ denotes the isotropic Gaussian kernel density, where c_i denotes the center of the ith patch, and y_k^l denotes the center of the lth particle in the kth frame. It means the distance between the patch and the candidate particle is considered, i.e., the patches far away from the center of the target will be assigned smaller weights, which can weaken the disturbance of the patches on the edge of the target.

According to the histogram intersection function [16, 20], the similarity function of the ith patch of the lth particle is defined as

$$L_{k,i}^l = \sum \min\left(\phi_{k,i}^l, \psi^i\right) \tag{8}$$

where $\phi_{k,i}^l$ and ψ^i denotes the sparse coefficient histograms of the candidate target and the target template, respectively. Template histogram is computed only once for each image sequence. Moreover, the comparison between the candidate and the template should be carried out under the same occlusion condition. Therefore, the template and the ith candidate share the same matrix o of occluded detection.

The likelihood of classification of the ith patch of the lth particle is defined as

$$M_{k,i}^l = \cos\angle\left(W\phi_{k,i}^l, \Gamma\right) \tag{9}$$

where $\phi_{k,i}^l$ is the sparse coefficient vector of the candidate patch. Γ denotes the base vector of target classification, i.e., $\Gamma = [1, 0]^T$, $\cos\angle(\alpha, \beta) = \frac{\alpha \cdot \beta}{|\alpha||\beta|}$ denotes the bearing of two vectors.

The bigger the number of patches belonging to the candidate particle is, the better the target appearance can be described. Because the selected patches may be from target templates or background templates. Therefore, if the patch belongs to the target, we should give it a bigger weight than that belong to the background.

2.4 Adaptive template update

In the tracking process, the appearance of the target often changes significantly due to the disturbance of illumination changes, occlusion, rotation, scale variation, and so on. Therefore, we need to update the template appropriately. However, if the template is updated too frequently by using new observations, the tracking results are easy to drift away from the target due to the accumulation of errors. Especially, when the target is occluded, the latest tracking result cannot describe the real target well, which will cause the later estimated targets to be lost. On the contrary, if tracking with fixed templates, it is prone to fail in dynamic scenes as it does not consider inevitable appearance change.

In this paper, we propose an improved template histogram update scheme by combining the histogram of the first frame and the latest estimated histogram with the variable μ, i.e.,

$$\hat{\psi}_n = \begin{cases} \mu\psi + (1-\mu)\hat{\phi}_n, & O_n < O_0 \\ \hat{\psi}_{n-1}, & \text{otherwise} \end{cases} \tag{10}$$

where $\mu = e^{-(1-\frac{O_n}{O_0})}$ denotes the weighting parameter, which can adaptively adjust the update template to adapt to the change of the target appearance. $\hat{\psi}_n$ denotes the update template histogram, ψ and $\hat{\phi}_n$ denote the template histogram of the first frame and the latest estimate, respectively. $O_n = \frac{\#\text{Patch}_{occ}}{\#\text{Patch}}$ denotes the occlusion degree of the current tracking results. $\#\text{Patch}_{occ}$ and #Patch denote the number of the occluded patches and the total patches. O_0 is a threshold of the degree of occlusion. Moreover, to avoid frequent template update, we detect the occluded state every five frames, i.e., we update the template every five frames.

During the update process, the first frame template and the newly arrived template are considered simultaneously. However, when the target is occluded, the arrived template usually cannot describe the real target effectively. Therefore, the weight μ of the arrived template should decrease at this time. Otherwise, the weight μ should increase due to the accurately estimate of the arrived template without other disturbance factors. In this paper, we set the parameter μ change with the reconstruction error. If O_n increases, which denotes the target

may be disturbed by some factors, such as illumination and occlusion, the arrived template may be inaccurate, hence the weight of the template should decrease, while the weight of the first frame template should increase.

3 Experiment results

To verify the effectiveness of the proposed algorithm, some challenging sequences from the public dataset of video target tracking [1] (http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html) are used to evaluate the performance of the proposed algorithm. The main challenging features of the data are described in Table 1, including the interference of occlusion, background clutter, illumination change, target rotation, scale change, motion blur, etc. The proposed algorithm is compared with eight state-of-the-art benchmark tracking algorithms, including multiple instance learning (MIL) [6], compressive tracking (CT) [21], robust fragments-based tracking (FRAG) [7], incremental visual tracking (IVT) [8], visual tracking decomposition (VTD) [9], L1 tracker using accelerated proximal gradient (L1APG) [14], multi-task sparse learning tracking (MTT) [15], and local sparse appearance model and K-selection (LSK) [22]. The experiments are implemented on computer with Intel Core 2.4 GHz, i7-4700HQ processor with 8 GB RAM. The software tool is MATLAB 2014a and the l_1 minimization problem is solved with the SPAMS package [23]. For each sequence, the location of the target is manually labeled in the first frame.

The learned low-dimensional dictionary consists of 50 positive templates and 50 background templates which are from the sampled templates by LC-KSVD dictionary learning. In the framework of PF, 100 candidate particles are sampled according to the same partition patch method, and the most similarity candidate particle is extracted as the estimated target. Set the threshold of the occlusion degree as $O_0 = 0.8$ in Eq. (10).

3.1 Qualitative evaluation

Figure 3 shows the tracking results of different algorithms when the target undergoes heavy occlusion, illumination variation, background clutter, rotation, scale change, fast motion, and motion blur.

3.1.1 Occlusion and illumination variation

In order to demonstrate the anti-occlusion and anti-illumination-variation performances of the proposed algorithm, some challenging video sequences are used in this experiment. Especially in (a) FaceOcc2 and (b) Woman sequences, the targets are heavily occluded or long-time partial occluded. However, the proposed algorithm can extract the targets accurately. The reason is that the local detection strategy for occlusion and illumination changes as well as the adaptive template update scheme are employed, which can easily describe and detect the variations of the local details of the targets and help to decrease the influence of the disturbances including occlusion, illumination change, rotation, etc. Moreover, the Gaussian kernel density of the patches is considered in the CKPF, which considers the global information of the local patches, improving the tracking performance. Taking the 181th, 273th, and 659th frames in FaceOcc2 sequences as examples, the target is occluded heavily by the book and the hat; the proposed algorithm has the highest tracking accuracy. In the 127th, 172th, and 495th frames in the Woman sequences, the target is partial occluded by the car and disturbed by the background clutters; some of the benchmark algorithms cannot estimate the target accurately with heavily position drift, while the proposed algorithm can successfully track the target throughout the entire sequences.

In (c) Shaking and (d) Singer1 sequences, there exists large illumination variation, and partial scale

Table 1 The features of the video sequences

Name	Occlusion	Blur	Scale change	Background clutter	Illumination change	Rotation	Fast motion	Deform
FaceOcc2	√				√	√		
Woman	√	√	√		√	√	√	√
Shaking			√	√	√	√		
Singer1	√		√		√	√		
Deer		√		√		√	√	
Board		√	√	√		√	√	
Trellis			√	√	√	√		
Walking2	√		√					
Girl	√		√			√		
Jumping		√					√	
Human8			√		√			√
Car4			√		√			

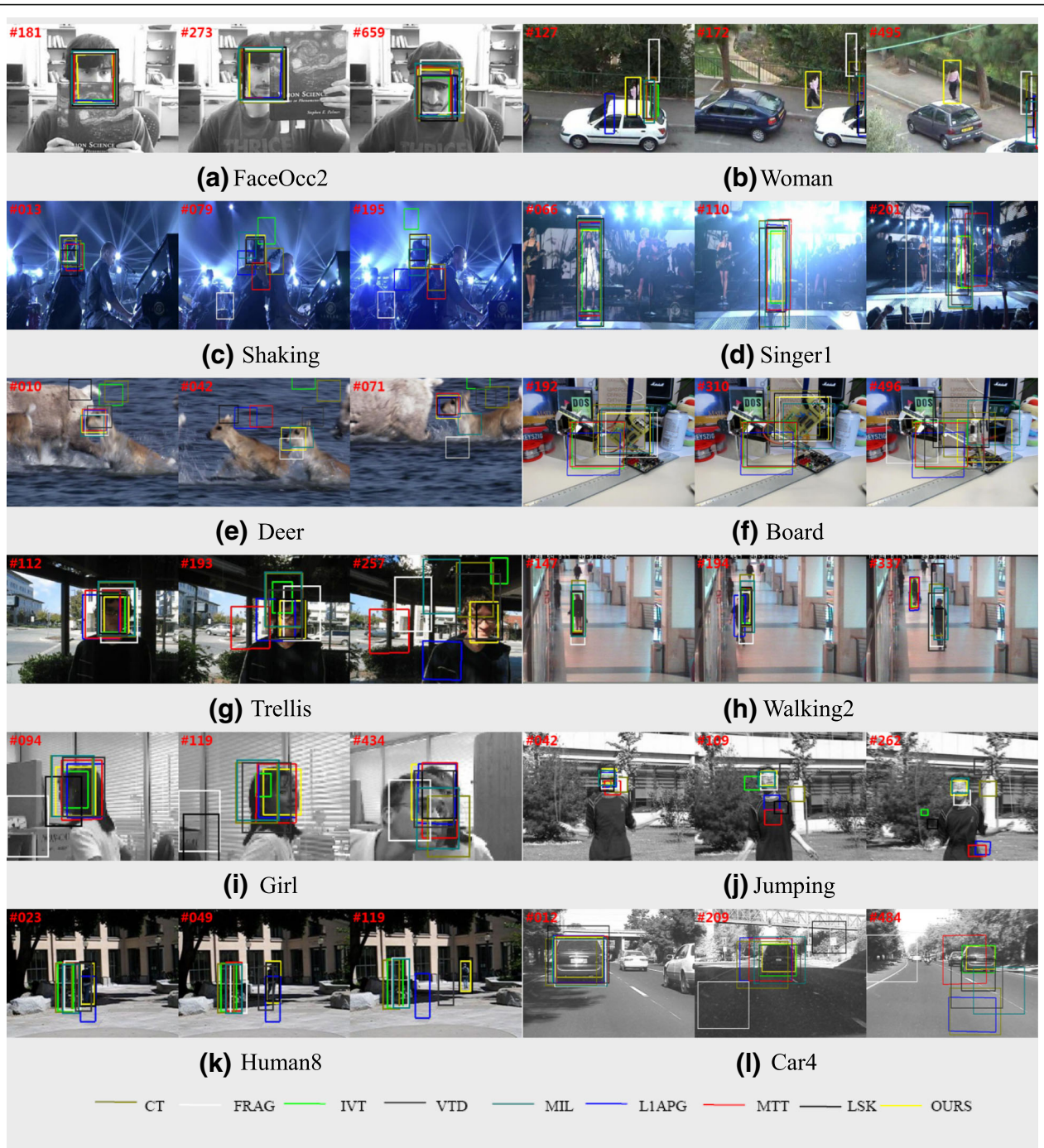

Fig. 3 Tracking results of different algorithms. **a** FaceOcc2. **b** Woman. **c** Shaking. **d** Singer1. **e** Deer **f** Board. **g** Trellis. **h** Walking2. **i** Girl. **j** Jumping. **k** Human8. **l** Car4

change, the benchmark algorithms FRAG, IVT, MTT, and CT cannot extract the target correctly following with heavily drift. LSK and MIL have good estimated results, but the proposed algorithm and the VTD approach have better tracking results. For the VTD algorithm, the observation model is decomposed into multiple basic observation models that can cover different specific target appearances, which can adapt to the illumination changes; however, it is hard to deal with the scale variation problem of the target while the proposed algorithm can do it adaptively. Therefore, in the Singer1 sequences, its tracking results are worse than those of the proposed algorithm due to the scale variation of the targets.

3.1.2 Background clutter

In the video sequences of (f) Board, (e) Deer, and (c) Shaking, the targets are disturbed by some background clutters, especially in Board sequences; the background is complex and there exists partial target rotation and fast motion. L1APG, MTT, and IVT cannot extract the target correctly due to the use of the fixed global model, while the proposed algorithm employs the local patch features to describe the details of the target, and the LC-KSVD method is introduced to learn dictionaries and train the classification parameters simultaneously, which can decrease the influence of the background disturbance. In the 42th frame of the Deer sequence, there is another deer in the background. Most of the algorithms have the results with largely drift due to the clutter disturbance. However, the proposed algorithm obtains an accurate result; the reason is that the set of background models is considered simultaneously and effectively updated in the tracking process.

3.1.3 Rotation and scale change

In (i) Girl and (f) Board sequences, there exists heavily target rotation. In the 94th and 119th frames of the Girl sequences, the girl turns around. It is clear that heavily drift exists in the results obtained by FRAG and LSK, while the proposed algorithm can adapt to the case of target rotation due to the use of the effectively update strategy, which considers the initial target model and the last estimate target model simultaneously. In the 434th frame of the Girl sequences, the face of the girl is occluded by the man and the scale makes a little change during the process of target rotation; the proposed algorithm also obtains a good tracking result. From the Board sequences, we can draw the same conclusions, in which the proposed algorithm has a good performance of target tracking under the scenario with target rotation and scale variation.

Moreover, in the Singer1 sequences, it is clear that the scale of the target changes heavily; the proposed algorithm can obtain accurate results, because the scale parameter s_k is estimated simultaneously in the implement process of CKPF.

3.1.4 Fast motion and motion blur

In (j) Jumping and (e) Deer sequences, there exists fast motion of the target and motion blur. For the Jumping sequences, L1APG, LSK, and MTT cannot extract the target correctly due to the motion blur, while the proposed algorithm has a good tracking result. In the 109th and 262th frames of the Jumping sequence, fast motion and motion blur make some of the benchmark algorithms have heavily drift results, while the proposed algorithm has good results. The reason is that the background templates are considered to restrain the influence of the background, and the updated positive template can adapt to the case with motion blur. From the Deer sequences, we can conclude the same conclusions.

3.2 Quantitative evaluation

Two evaluation criteria are employed to quantitatively assess the performance of the proposed algorithm. One is average center location error (ACLE), and the other is tracking success rate (SR). Figure 4 shows the relative position error (in pixels) between the center and the tracking results. ACE is defined as the average relative position error. Assume the tracking result is R_r, and the ground truth is R_g, then SR is defined as $\Upsilon = (R_r \cup R_g)/(R_r \cup R_g)$. Tables 2 and 3 give values of ACLE and SR for different tracking algorithms.

As can be seen from Fig. 4, the proposed algorithm has a better performance than those of the benchmark algorithms. The tracking result of each frame is accurate and the curve of the error is stable without high changing. While part of the benchmark algorithms are instable, and have big errors between some frames due to different disturbances.

From Tables 2 and 3, it is clear that the proposed algorithm can adapt to most of the video sequences with the best and second best results except the (i) Girl sequences. The performance of the proposed algorithm can be attributed to the detailed description of the local patches by the LC-KSVD dictionary learning and adaptive template update scheme. Moreover, the Gaussian kernel density of the patches as the global information is considered in CKPF. The algorithm of VTD can also adapt to the scenarios with illumination change and lightly occlusion (e.g., Shaking and Singer1); the reason is that the appearance change is considered in the target template, but its performance decreases when the rotation and the motion blur happen on the targets (e.g., Deer, Board, and Jumping). L1APG has a good performance on the Girl sequence; the reason is that the last tracking result is used directly as the updated template, which can effectively adapt to the Girl sequence with the turn of the girl. However, it cannot extract the target correctly due to the motion blur and illumination variation, such as in (f) Board, (j) Jumping, (c) Shaking, and (l) Car4 sequences. For the Girl sequences, the tracking result of the proposed algorithm is not the best, but it is only slightly below the L1APG and MTT algorithms.

3.3 Discussion of adaptive parameter μ

To verify the effectiveness of the adaptive template update scheme, two special challenging sequences, the first 200 frames of FaceOcce2 and the first 170 frames of Woman with big variance of appearance, are chosen in this experiment. The tracking results with different

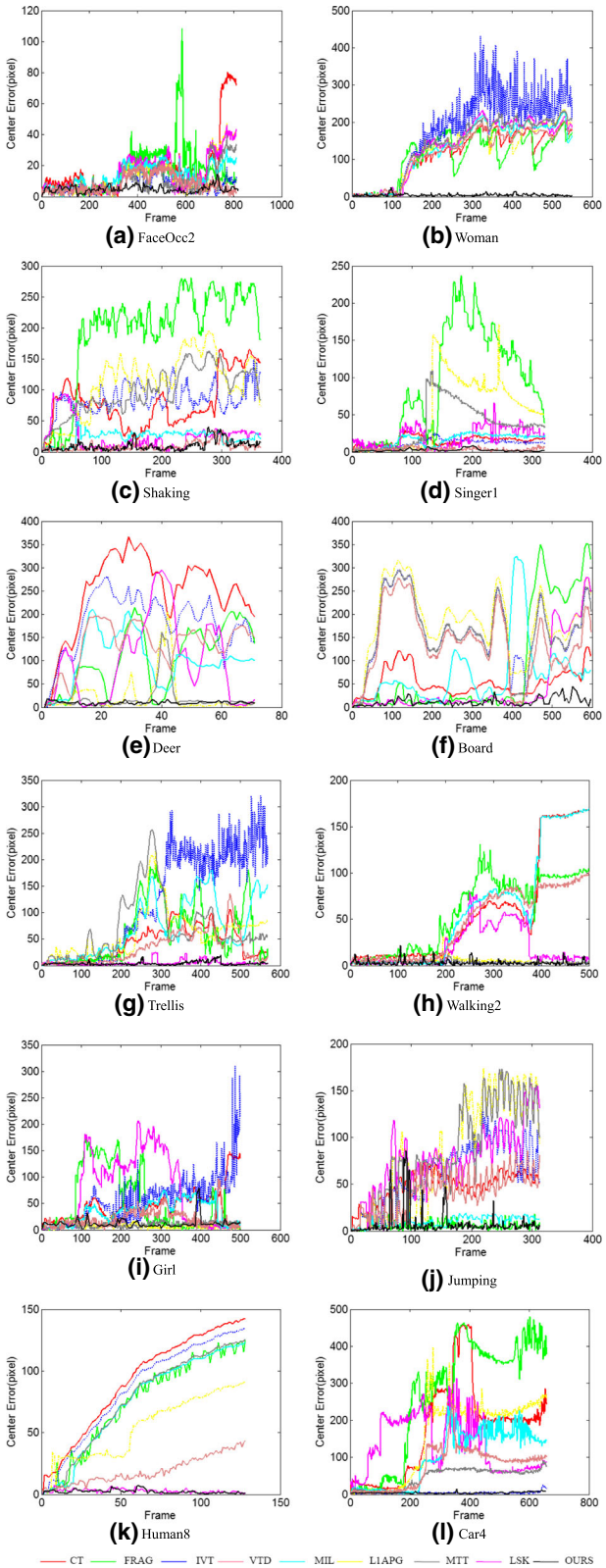

Fig. 4 Position errors (in pixel) between the center and the tracking results. **a** FaceOcc2. **b** Woman. **c** Shaking. **d** Singer1. **e** Deer. **f** Board. **g** Trellis. **h** Walking2. **i** Girl. **j** Jumping. **k** Human8. **l** Car4

Table 2 Average center location error (in pixel). The best and second best results are shown in italic and bold

	IVT	MIL	VTD	FRAG	CT	L1APG	MTT	LSK	OURS
FaceOcc2	**6.9**	13.6	8.1	15.7	19.3	12.9	10.2	14.7	*4.88*
Woman	172.6	126.1	117.6	**109.7**	113.7	126.7	134.8	131.6	*4.43*
Shaking	85.4	24.0	*9.0*	192.1	80.0	109.7	97.3	25.5	**10.02**
Singer1	11.5	16.1	**4.0**	91.5	15.4	53.1	35.9	21.2	*2.48*
Deer	182.8	100.7	134.8	105.1	246.4	24.2	**19.1**	98.8	*9.95*
Board	162.2	71.5	137.7	84.5	52.8	184.4	159.2	**45.4**	*12.08*
Trellis	119.57	71.47	32.25	59.51	41.69	62.30	68.99	**4.70**	*3.85*
Walking2	**3.04**	60.65	46.25	57.53	58.53	4.52	3.48	18.95	*2.84*
Girl	56.6	34.1	21.5	51.7	47.1	*6.9*	**10.4**	73.2	11.8
Jumping	61.3	10.0	41.4	*5.6*	47.7	83.3	84.1	74.6	**7.74**
Human8	85.96	74.95	19.00	74.83	92.14	54.17	76.42	**2.74**	*2.18*
Car4	**4.08**	101.55	73.99	263.1	172.05	153.98	45.25	133.23	*3.89*

constant values (e.g., 0.1, 0.4, 0.7, and 0.9) of the weighting parameter μ of Eq. (10) are compared to those with adaptive parameter value, and these are demonstrated in Table 4.

As can be seen that there are different values of ACLEs and SRs by choosing different constant values of μ, and smaller value of μ (e.g., 0.1) gets higher accuracy for the first 200 frames FaceOcce2 sequences, while bigger value of μ (e.g., 0.9) gets higher accuracy for the first 170 frames Woman sequences. The reason is that the variations of the target appearance are small during the 1st frame to 140th frame of FaceOcce2 sequences, and the updated templates mainly rely on the latest templates. But the target appearances are severely occluded between 141st and 190th fames; the updated templates more rely on the template of the first frame. Therefore, it is noted that the differences of the tracking accuracy are small with different values of μ for this sequences.

Table 3 Success rate. The best and second best results are shown in italic and bold

	IVT	MIL	VTD	FRAG	CT	L1APG	MTT	LSK	OURS
FaceOcc2	0.73	0.68	0.74	0.66	0.61	0.68	**0.75**	0.64	*0.82*
Woman	0.16	0.17	0.16	0.16	0.14	0.17	**0.18**	0.17	*0.74*
Shaking	0.03	0.43	*0.71*	0.08	0.10	0.08	0.04	0.46	**0.65**
Singer1	**0.59**	0.39	0.53	0.23	0.38	0.32	0.37	0.37	*0.87*
Deer	0.03	0.13	0.06	0.17	0.04	*0.62*	*0.62*	0.27	**0.6**
Board	0.15	0.46	0.22	0.55	0.50	0.11	0.16	**0.65**	*0.83*
Trellis	0.25	0.25	0.46	0.29	0.34	0.20	0.22	**0.66**	*0.71*
Walking2	0.76	0.29	0.33	0.28	0.27	**0.78**	*0.81*	0.47	0.75
Girl	0.17	0.40	0.56	0.45	0.31	*0.74*	**0.67**	0.31	0.62
Jumping	0.12	0.53	0.13	*0.68*	0.05	0.15	0.10	0.07	**0.66**
Human8	0.06	0.12	0.30	0.10	0.04	0.16	0.10	**0.69**	*0.74*
Car4	*0.88*	0.26	0.37	0.19	0.22	0.26	0.45	0.15	*0.89*

But for the Woman sequences, the target appearances are slightly disturbed by the background clutters between 36th and 170th frames, and there only exists partial occlusion between 106th and 165th frames. Therefore, most of the updated templates mainly rely on the latest frame templates, and the bigger value of μ gets better results. While for the proposed algorithm with adaptive weight parameter, it is clear that it can obtain an ideal tracking result without manually setting the parameter values.

4 Conclusion

In this paper, we present an adaptive visual tracking algorithm based on CKPF. The template sets constructed by the local patch features from both foreground and background of the target are used to learn the dictionaries simultaneously. The low-dimensional dictionary and target classification parameters are trained by using the LC-KSVD dictionary learning. To robustly decide the final tracking states, an adaptive template update scheme is designed, and the classification information, the target candidate histogram, and the Gaussian kernel density are merged to form CKPF. The effectiveness of the proposed algorithm is experimentally demonstrated by comparing with 8 state-of-the-art trackers on 12 challenging video sequences, and experimental results show that the proposed

Table 4 Discussion of constant and adaptive parameter μ. The best results are shown in italic

	Evaluation criteria	μ				
		0.1	0.4	0.7	0.9	Adaptive
FaceOcc2 1~200 frames	ACLE	*4.48*	4.53	4.65	4.81	4.59
	SR	*0.85*	0.85	0.84	0.84	0.84
Woman 1~170 frames	ACLE	15.66	5.80	4.39	4.32	*2.87*
	SR	0.58	0.80	0.82	0.82	*0.84*

algorithm has a better tracking performance than some benchmark methods in the scenarios with the interference of occlusion, background clutter, illumination change, target rotation, and scale change. However, the computation cost is high; in the future, we would like to improve the computational efficiency by considering the reverse-low-rank representation scheme [24], and some optimal particle pruning schemes.

Abbreviations

APG: Accelerated proximal gradient; ASCH: Anti-occlusion sparse coefficient histograms; CKPF: Classified-patch kernel particle filter; CP: Classification parameters; CT: Compressive tracking; GKD: Gaussian kernel density; IVT: Incremental learning visual tracking; LC-KSVD: Label-consistent K-singular value decomposition; LS: Least squares; LSK: Local sparse appearance model and K-selection; MIL: Multiple instance learning; ML: Maximum likelihood; MTT: Multi-task tracking; PF: Particle filtering; VTD: Visual tracking decomposition; VTT: Visual target tracking

Acknowledgments

The authors would like to thank the Editor and anonymous reviewers for their constructive suggestion.

Funding

Natural Science Foundation of Jiangsu Province (Nos. BK20181340, BK20130154), National Natural Science Foundation of China (Nos. 61305017, 61772237), and The Cyber-Physical Systems program of the U.S. National Science Foundation (CNS 1329481).

Availability of data and materials

All data and material are available.

Authors' contributions

JY initiated the project. GZ, JY, and JL designed the algorithms, performed the experiments, and drafted the manuscript. WW and YH participated in the proposed method and analyzed the experiment results. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]College of Information Engineering, Chang'an University, Xi'an 710064, China. [2]School of Computer Science and Technology, Baoji University of Arts and Science, Baoji 721076, China. [3]School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China. [4]Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI 53706, USA.

Received: 31 August 2018 Accepted: 6 January 2019
Published online: 24 January 2019

References

1. Y. Wu, J. Lim, M.H. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015).
2. M. Kristan, J. Matas, A. Leonardis, et al., in *Proceedings of the IEEE international conference on computer vision workshops*. The visual object tracking vot2015 challenge results (2015), pp. 1–23.
3. H. Fan, J. Xiang, Robust visual tracking with multitask joint dictionary learning. IEEE Trans. Circuits Syst. Video Technol. **27**(5), 1018–1030 (2017).
4. H. Li, Y. Li, F. Porikli, Deep track: learning discriminative feature representations online for robust visual tracking. IEEE Trans. Image Process. **25**(4), 1834–1848 (2016).
5. X. Jia, H.C. Lu, M.H. Yang, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking via adaptive structural local sparse appearance model (IEEE Computer Society Press, Los Alamitos, 2012), pp. 1822–1829.
6. B. Babenko, M.H. Yang, S. Belongie, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking with online multiple instance learning (IEEE Computer Society Press, Los Alamitos, 2009), pp. 983–990.
7. A. Adam, E. Rivlin, I. Shimshoni, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Robust fragments-based tracking using the integral histogram (IEEE Computer Society Press, Los Alamitos, 2006), pp. 798–805.
8. D.A. Ross, J. Lim, R.S. Lin, et al., Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008).
9. J. Kwon, K.M. Lee, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking decomposition (IEEE Computer Society Press, Los Alamitos, 2010), pp. 1269–1276.
10. X. Mei, H.B. Ling, in *Proceedings of IEEE 12th International Conference on Computer Vision*. Robust visual tracking using L1 minimization (IEEE Computer Society Press, Los Alamitos, 2009), pp. 1436–1443.
11. X. Mei, H.B. Ling, Y. Wu, et al., in *Proceedings of IEEE conference on computer vision and pattern recognition*. Minimum error bounded efficient L1 tracker with occlusion detection (IEEE Computer Society Press, Los Alamitos, 2011), pp. 1257–1264.
12. M.S. Arulampalam, S. Maskell, N. Gordon, et al., A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002).
13. S.P. Zhang, H.X. Yao, X. Sun, et al., Sparse coding based visual tracking: Review and experimental comparison. Pattern Recogn. **46**(7), 1772–1788 (2013).
14. C.L. Bao, Y. Wu, H.B. Ling, et al., in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Real time robust L1 tracker using accelerated proximal gradient approach (IEEE Computer Society Press, Los Alamitos, 2012), pp. 1830–1837.
15. T.Z. Zhang, B. Ghanem, S. Liu, et al., in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Robust visual tracking via multi-task sparse learning (IEEE Computer Society Press, Los Alamitos, 2012), pp. 2042–2049.
16. W. Zhong, H.C. Lu, M.H. Yang, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Robust object tracking via sparsity-based collaborative model (IEEE Computer Society Press, Los Alamitos, 2012), pp. 1838–1845.
17. Z.L. Jiang, Z. Lin, L.S. Davis, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Learning a discriminative dictionary for sparse coding via label consistent k-svd (IEEE Computer Society Press, Los Alamitos, 2011), pp. 1697–1704.
18. Z.L. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning a discriminative dictionary for recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2651–2664 (2013).
19. M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing Overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006).
20. J.X. Wu, J.M. Rehg, in *Proceedings of IEEE 12th International Conference on Computer Vision*. Beyond the Euclidean distance: creating effective visual codebooks using the histogram intersection kernel (IEEE Computer Society Press, Los Alamitos, 2009), pp. 630–637.
21. K.H. Zhang, L. Zhang, M.H. Yang, in *Proceedings of the 11th European Conference on Computer Vision*. Real-time compressive tracking (IEEE Computer Society Press, Los Alamitos, 2012), pp. 864–877.
22. B.Y. Liu, J.Z. Huang, L. Yang, et al., in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Robust tracking using local sparse appearance model and K-selection (IEEE Computer Society Press, Los Alamitos, 2011), pp. 1313–1320.
23. J. Mairal, F. Bach, J. Ponce, et al., Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res. **11**(1), 19–60 (2010).
24. Y. Yang, W. Hu, Y. Xie, et al., Temporal restricted visual tracking via reverse-low-rank sparse learning. IEEE Trans. Cybern. **47**(2), 485–498 (2017).