Robust Online Object Tracking with a Structured Sparse Representation Model

Chunjuan Bo[1,2], Dong Wang[1]
[1] School of Information and Communication Engineering, Dalian University of Technology
Dalian, Liaoning 116024 - China
[e-mail: wdice@dlut.edu.cn]
[2] College of Electromechanical Engineering, Nationalities University, Dalian, Liaoning 116600 - China
[e-mail: bcj@dlnu.edu.cn]
*Corresponding author: Dong Wang

Received December 3, 2015; revised January 16, 2016; accepted April 7, 2016; published May 31, 2016

Abstract

As one of the most important issues in computer vision and image processing, online object tracking plays a key role in numerous areas of research and in many real applications. In this study, we present a novel tracking method based on the proposed structured sparse representation model, in which the tracked object is assumed to be sparsely represented by a set of object and background templates. The contributions of this work are threefold. First, the structure information of all the candidate samples is utilized by a joint sparse representation model, where the representation coefficients of these candidates are promoted to share the same sparse patterns. This representation model can be effectively solved by the simultaneous orthogonal matching pursuit method. In addition, we develop a tracking algorithm based on the proposed representation model, a discriminative candidate selection scheme, and a simple model updating method. Finally, we conduct numerous experiments on several challenging video clips to evaluate the proposed tracker in comparison with various state-of-the-art tracking algorithms. Both qualitative and quantitative evaluations on a number of challenging video clips show that our tracker achieves better performance than the other state-of-the-art methods.

Keywords: Object tracking, sparse representation, simultaneous orthogonal matching pursuit (SOMP)

This research was supported in part by the Natural Science Foundation of China under grant no. 61502070, and in part by the Fundamental Research Funds for Central Universities under grant no. DC201501010401.

http://dx.doi.org/10.3837/tiis.2016.05.021 **ISSN : 1976-7277**

1. Introduction

Online object tracking is a fundamental and interesting issue in the fields of video processing and computer vision, and has many applications in real-world scenarios, including human–computer interface, video surveillance, intelligent traffic, and augmented reality. Although several processes have been introduced and numerous effective tracking algorithms have been developed [1][2], designing a robust and efficient tracker remains extremely difficult because of many challenging factors. These factors mainly include heavy occlusion, illumination variation, pose change, background clutter, and motion blur.

An online object-tracking algorithm generally consists of two basic components: motion model and appearance model. The motion model focuses on depicting the states of the tracked object over time, and thus, generate a set of possible candidate states in each frame. In many classical or recent tracking algorithms, the Kalman filtering [3] and particle filtering [4][5] techniques are the commonly used motion models. By contrast, the appearance model aims to represent the appearance of the tracked object (and its surrounding background) to evaluate the likelihood of each candidate in the current frame. In view of the appearance model, existing tracking methods can be mainly categorized into algorithms based on template matching [3][6][7][8], online classifiers [9][10][11][12][13][14][15], and representation models [16][17][18][19][20]. Tracking algorithms based on template matching frequently use a single template [3][6][8] or multiple templates [7] to depict the tracked object; they cast the tracking problem as finding the best candidate image patch with the highest similarity or the smallest distance. Tracking algorithms based on online classifiers are also called discriminative tracking methods; they aim to distinguish tracked objects from their surrounding backgrounds and update the classification models online to capture appearance changes from both foregrounds and backgrounds. Therefore, several existing classification techniques can promote the progress of tracking algorithms, such as support vector machine [10][15], boosting [9]–[11], and multiple instance learning (MIL) [12].

Trackers based on representation models generally include two main categories: subspace-based trackers and sparse representation-based trackers. In 2008, Ross *et al.* [16] proposed an incremental visual tracking (IVT) method that adopted an incremental principle component analysis technique to model the appearance of a tracked object online. Li *et al.* [17] introduced the concept of manifold learning into the IVT method and adopted log-Euclidean distance to improve the robustness of the tracker. Although these methods achieve good tracking performance when the tracked object experiences illumination variation and pose change, they are less effective in dealing with other challenging factors such partial occlusion and background clutter. Motivated by the success of sparse representation, Mei *et al.* [18] developed a novel L1 tracker that adopted a set of objects and trivial templates to sparsely approximate a tracked object, in which the L1 regularization term promoted a sparse solution. Although the L1 tracker [18] explicitly introduces theory of sparse representation into the tracking field and model outliers (e.g., partial occlusion) by using trivial templates, its performance is unsatisfactory because of two reasons. First, the L1 tracker has to adopt low-resolution image patches to balance speed and accuracy because this tracker requires solving numerous complicated L1 minimization problems in each frame. Second, the L1 tracker uses a simple model updating method, which causes the tracker to drift easily because of inappropriate updates. Many researchers have attempted to improve the L1 tracker by combining subspace and sparse representation models [19], modeling the relationships among

different candidates [20], and adopting different optimization techniques [21]. In the present study, we propose a novel online object-tracking algorithm based on sparse representation. The contributions of this work are threefold.

(1) We propose a structured sparse representation model to represent all candidate samples through a set of object and background templates. This representation model can be effectively solved using the simultaneous orthogonal matching pursuit (SOMP) method.

(2) On the basis of the particle filter framework, we develop a tracking framework using the proposed structured sparse representation model with a discriminative candidate selection scheme and a simple updating manner.

(3) We conduct several experiments on nine challenging image sequences to investigate the effects of the key parameters of our tracker and to compare the proposed tracking algorithm with eight state-of-the-art trackers. The experimental results demonstrate that the proposed tracker achieves good performance compared with the other methods.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the background of the study and related works. In Section 3, we present the proposed tracking framework based on the structured sparse representation model. In Section 4, we report the experimental results of the proposed tracker and compare the tracker with other state-of-the-art tracking methods. Finally, we draw our conclusions in Section 5.

2. Background and Related Works

2.1 Particle filter

The particle filter [4][5][16] is a common framework for solving the tracking problem because this algorithm can be regarded as a typical dynamic state inference problem. To ensure a self-contained approach in this study, we briefly introduce several fundamental concepts of the particle filter technique. The particle filter is a Bayesian sequential importance sampling algorithm that can estimate the posterior distribution of state variables for a given dynamic system using a finite set of weighted samples. Notably, the particle filter provides a unified framework for estimating and propagating the posterior probability density function of state variables regardless of the underlying distribution.

For the tracking problem, we adopt the symbol $\mathbf{z}_t$ to denote the state variable that describes the affine motion parameters of the target, and $\mathbf{y}_t$ to denote its corresponding observation vector, i.e., the extracted image feature related to state $\mathbf{z}_t$ (t is the frame index). The prediction and updating steps recursively estimate the posterior probability of the tracked state based on the following rules:

$$p(\mathbf{z}_t \mid \mathbf{y}_{1:t-1}) = \int p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) p(\mathbf{z}_{t-1} \mid \mathbf{y}_{1:t-1}) d\mathbf{z}_{t-1}, \tag{1}$$

$$p(\mathbf{z}_t \mid \mathbf{y}_{1:t}) = p(\mathbf{y}_t \mid \mathbf{z}_t) p(\mathbf{z}_{t-1} \mid \mathbf{y}_{1:t-1}) / p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1}), \tag{2}$$

where $\mathbf{z}_{1:t} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_t]$ denotes all the available states up to frame t, and $\mathbf{y}_{1:t} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t]$ indicates their corresponding observation samples. In the tracking problem, $p(\mathbf{z}_t \mid \mathbf{z}_{t-1})$ depicts the state transition between two consecutive frames, which is frequently called the motion model. $p(\mathbf{y}_t \mid \mathbf{z}_t)$ denotes the appearance model that aims to evaluate the observation likelihood of each observation sample being similar to the tracked object.

On the basis of the particle filter framework, the posterior distribution $p(\mathbf{z}_t \mid \mathbf{y}_{1:t})$ can be approximated using N weighted particles $\{\mathbf{z}_t^i, w_t^i\}\big|_{i=1}^N$ drawn from an importance distribution $q(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t})$, where w_t^i denotes the weight of particle $\mathbf{z}_t^i$. The weights of the particles are updated frame-by-frame based on Equation (3):

$$w_t^i = w_{t-1}^i \left[p(\mathbf{y}_t^i \mid \mathbf{z}_t^i)\, p(\mathbf{z}_t^i \mid \mathbf{z}_{t-1}^i) \big/ q(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}) \right]. \tag{3}$$

Similar to that in reference [16], we use $q(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{z}_t^i \mid \mathbf{z}_{t-1}^i)$, which is assumed to follow a Gaussian distribution. In particular, we adopt the six parameters, i.e., $\mathbf{z}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$, of affine transform to represent the state of the tracked object, where parameters $x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t$ denote vertical translation, horizontal translation, rotation angle, scale, aspect ratio, and skew, respectively. Moreover, state transition is formulated via a simple random walking process, i.e., $p(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \mathbf{z}_{t-1}, \boldsymbol{\psi})$, where $\boldsymbol{\psi}$ is a diagonal covariance matrix, and the covariances of which are denoted as $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$. Finally, the optimal state can be obtained via $\mathbf{z}_t^* = \sum_{i=1}^{N} \mathbf{z}_t^i$ (the optimal state represents the best accurate motion parameter in the t-th frame).

2.2 Sparse representation-based object tracking

At present, sparse representation has been extensively investigated and applied to solve many real-life problems in numerous research fields, such as pattern recognition, image processing, and computer vision [22]. Sparse representation-based object-tracking algorithms have also been shown to achieve significant performance compared with traditional trackers. In 2009, motivated by the success of the sparse representation-based face recognition system [23], Mei *et al.* [18] proposed a novel L1 tracker (denoted as the original L1 tracker) by assuming that a tracked object could be sparsely represented by a set of target templates and trivial templates; the target templates would describe the appearance change of the tracked object, whereas the trivial templates would deal with possible occlusion conditions. This sparse representation process is achieved by solving an L1 minimization problem. However, the original L1 tracker has two main drawbacks. First, its computational complexity is extremely high because of the complicated L1 minimization. Second, this tracker does not utilize rich and redundant image properties because low-solution image patches (12×15 pixels [18]) are adopted to balance speed and accuracy.

To improve the performance of the original L1 tracker, numerous researchers have conducted studies based on different perspectives. Mei *et al.* [24] presented an efficient L1 tracker with a novel minimum error bound scheme. In this scheme, an error bound can be estimated quickly using an ordinary least squares regression and then used to select effective candidates. Since then, several methods have been presented to improve the original L1 tracker in terms of both speed and accuracy, such as introducing dimension reduction techniques [25], adopting the accelerated proximal gradient algorithm [21], modeling similarities among different candidates [20], and using orthogonal basis vectors to replace raw pixel templates [19]. In the current work, our tracking algorithm is designed based on sparse representation, which is motivated by the success of the aforementioned trackers.

3. Object Tracking based on a Structured Sparse Representation Model

3.1 Feature extraction

Notably, several unexpected abnormal noises (such as local illumination variation and partial occlusions) may occur along with the appearance of a tracked object during the tracking process. These noises will inevitably affect the performance of an object-tracking algorithm. In this work, we address this issue by adopting fragment-based feature representation with local normalization. The feature extraction process is illustrated in **Fig. 1**. First, an observation image patch $\mathbf{Y}$ is divided into $M \times M$ fragments. For each fragment, the grayscale information is extracted and vectorized into a column vector. Second, the feature vectors are normalized to enable each of them to have an L2-norm unit (i.e., $\mathbf{y}_i \leftarrow \mathbf{y}_i / \|\mathbf{y}_i\|_2$, where index i denotes the i-th fragment). Finally, the normalized feature vectors are concatenated into a holistic feature vector $\mathbf{y}$.

$$\mathbf{y} = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_{M\times M}^T\right]^T \tag{4}$$

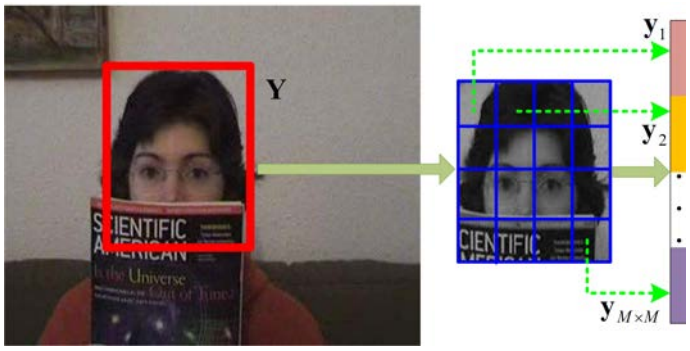

Fig. 1. Illustration of our feature extraction process.

3.2 Structured sparse representation model

In the t-th frame, we crop out the corresponding image patch for each particle $\mathbf{z}_t^i$ and extract its gray-level locally normalized feature $\mathbf{y}_t^i$ (the particles are randomly generated according to the motion model introduced in Section 2.1). Then, we construct the candidate set as $\mathbf{Y}_t = \left[\mathbf{y}_t^1, \mathbf{y}_t^2, ..., \mathbf{y}_t^N\right]$. In this section, we omit frame index t for clarity, i.e., $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\right]$, where column atom $\mathbf{y}_i$ denotes the feature of the i-th candidate. Motivated by [18], we assume that the tracked target can be sparsely represented by a series of templates $\mathbf{A} = \left[\mathbf{A}^+, \mathbf{A}^-\right]$, where $\mathbf{A}^+ = \left[\mathbf{a}_1^+, \mathbf{a}_2^+, ..., \mathbf{a}_{N_p}^+\right]$ and $\mathbf{A}^- = \left[\mathbf{a}_1^-, \mathbf{a}_2^-, ..., \mathbf{a}_{N_n}^-\right]$ represent the positive and negative templates, respectively (N_p is the number of positive samples, and N_n is the number of negative samples). The construction and updating of these templates are introduced in Section 3.4. Therefore, from the perspective of sparse representation, the approximations of all N candidates are illustrated in Equation (5). In this work, we adopt L0 norm ($\|.\|_0$, the number of nonzero numbers) to indicate the sparsity level.

$$\begin{cases} \mathbf{y}_1 \approx \mathbf{A}\mathbf{x}_1, & \|\mathbf{x}_1\|_0 \le T_0 \\ \mathbf{y}_2 \approx \mathbf{A}\mathbf{x}_2, & \|\mathbf{x}_2\|_0 \le T_0 \\ \ldots & \\ \mathbf{y}_N \approx \mathbf{A}\mathbf{x}_N, & \|\mathbf{x}_N\|_0 \le T_0 \end{cases}, \tag{5}$$

which can be rewritten in matrix form as follows:

$$\mathbf{Y} \approx \mathbf{A}\mathbf{X}, \quad \|\mathbf{X}\|_0 \le NT_0 . \tag{6}$$

However, this representation model regards each candidate individually and fails to consider the relationships among different candidates. To model the relationships among different candidates, we adopt a structured sparse representation model in this work by manipulating the coefficients of different candidates to share similar sparse patterns. The representation process can be modified into a problem, i.e., Equation (7), and an illustration of which is provided in **Fig. 2**.

$$\mathbf{Y} \approx \mathbf{A}\mathbf{X}, \quad \|\mathbf{X}\|_{row,0} \le T_0 ; \tag{7}$$

where notion $\|\mathbf{X}\|_{row,0}$ denotes the number of nonzero rows of $\mathbf{X}$. The solution of $\mathbf{X}$ can be obtained by solving the following optimization problem:

$$\begin{aligned} &\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \\ &s.t. \|\mathbf{X}\|_{row,0} \le T_0 \end{aligned} . \tag{8}$$

Notably, this optimization problem, i.e., Equation (8), can be solved via SOMP [26], which is a greedy algorithm that iteratively recovers the common support set. At each step, the column of the templates that can approximate all the residual vectors is selected and integrated into the support set. The iteration process terminates when the desired sparsity level is achieved. Detailed information on the SOMP method is provided in **Algorithm 1**. After obtaining the optimal coefficients, the likelihood value of each candidate $\mathbf{z}_t^i$ can be obtained using the following equation. The intuitive concept of Equation (9) is as follows: a good candidate should have a small reconstruction error based on positive samples and a large reconstruction error based on negative samples. Therefore, the proposed likelihood measure can utilize both foreground and background information for robust tracking.

$$p\left(\mathbf{y}_t^i \mid \mathbf{z}_t^i\right) = \exp\left[-\left(\left\|\mathbf{y}_t^i - \mathbf{A}_t^+ \hat{\mathbf{x}}_i^+\right\|_2^2 - \left\|\mathbf{y}_t^i - \mathbf{A}_t^- \hat{\mathbf{x}}_i^-\right\|_2^2\right)\right] \tag{9}$$

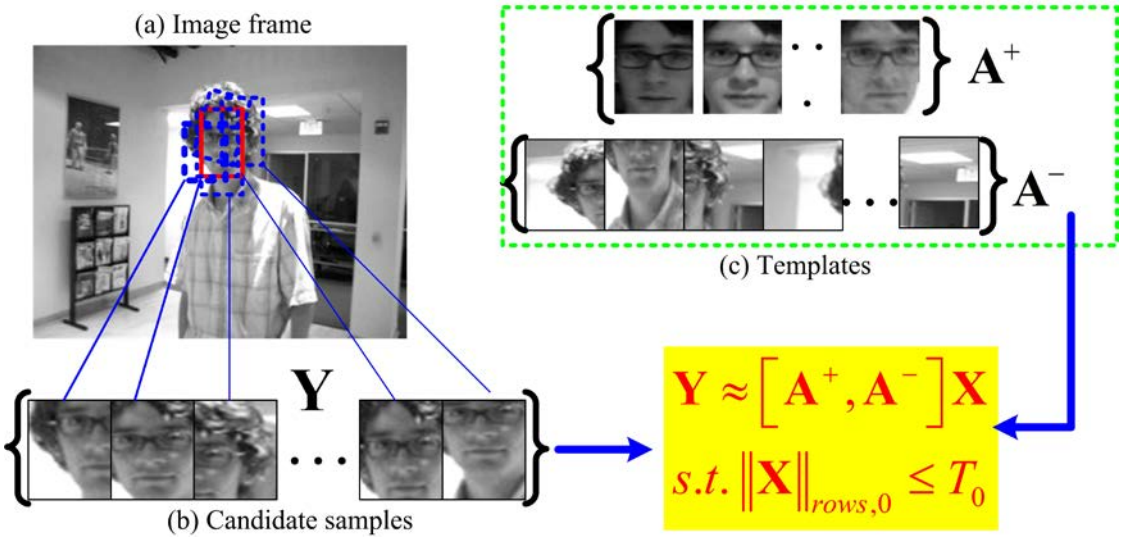

Fig. 2. Illustration of the proposed representation model.

Algorithm 1. Flowchart of the SOMP method.

Input: Candidate sample matrix $\mathbf{Y}$, template set $\mathbf{A}$, and sparsity level T_0

Step 1: Initialize residual matrix $\mathbf{R} = \mathbf{Y}$, index set $\mathbf{\Omega} = \phi$.

Step 2: **For** $i = 1$ *to* T_0 **do**

(1) Find the index of the atom that best approximates all residuals:

$$index = \arg\max_j \left\| \mathbf{R}^T \mathbf{a}_j \right\|_F^2, \; j = 1, 2, ..., \left(N_p + N_n \right).$$

(2) Update index set $\mathbf{\Omega} \leftarrow \mathbf{\Omega} \bigcup \{index\}$.

(3) Compute the coefficient matrix $\hat{\mathbf{X}} = \left(\mathbf{A}_{\mathbf{\Omega}}^T \mathbf{A}_{\mathbf{\Omega}} \right)^{-1} \mathbf{A}_{\mathbf{\Omega}}^T \mathbf{Y}$, where $\mathbf{A}_{\mathbf{\Omega}}$ consists of the columns indexed in $\mathbf{\Omega}$.

(4) Update residual matrix $\mathbf{R} = \mathbf{Y} - \mathbf{A}_{\mathbf{\Omega}} \hat{\mathbf{X}}$.

End For

Output: Index set $\mathbf{\Omega}$, the sparse representation coefficient matrix $\hat{\mathbf{X}}$ whose nonzero rows indexed by $\mathbf{\Omega}$ are the T_0 rows of matrix $\left(\mathbf{A}_{\mathbf{\Omega}}^T \mathbf{A}_{\mathbf{\Omega}} \right)^{-1} \mathbf{A}_{\mathbf{\Omega}}^T \mathbf{Y}$.

3.3 Discriminative candidate selection

To improve both the effectiveness and efficiency of our tracker, we present a discriminative candidate selection scheme to prune the original candidate sample matrix $\mathbf{Y} \in \mathbb{R}^{D \times N}$ into a small pruned sample matrix $\mathbf{Y}' \in \mathbb{R}^{D \times K}$, where D is the dimension of each sample and K is the number of selected candidates ($K < N$). The pseudo-code of the proposed scheme is provided in **Algorithm 2**. The basic concept of this algorithm is as follows: a good candidate

sample has a smaller distance to the positive template and a large distance to the negative template. In this work, we adopt the nearest neighbor distance to measure these two distances, i.e., the distance between $\mathbf{y}_i$ and the positive template set $\mathbf{A}^+$ is defined as

$$d_i^+ = \min_j \left\| \mathbf{y}_i - \mathbf{a}_j^+ \right\|_2^2, j = 1,2,...,N_p, \tag{10}$$

whereas the distance between $\mathbf{y}_i$ and the negative template set $\mathbf{A}^-$ is calculated by

$$d_i^- = \min_j \left\| \mathbf{y}_i - \mathbf{a}_j^- \right\|_2^2, j = 1,2,...,N_n. \tag{11}$$

Algorithm 2. Flowchart of the proposed candidate selection scheme.

Input: Candidate sample matrix $\mathbf{Y}$, template set $\mathbf{A}$, and selected number K

Step 1: Calculate a distance vector $\mathbf{d}$ for candidate selection.

For $i = 1$ *to* N **do**

$d_i^+ = \min_j \left\| \mathbf{y}_i - \mathbf{a}_j^+ \right\|_2^2, j = 1,2,...,N_p$ and $d_i^- = \min_j \left\| \mathbf{y}_i - \mathbf{a}_j^- \right\|_2^2, j = 1,2,...,N_n$

$d_i = d_i^+ - d_i^-$

End For

Step 2: Sort distances in ascending order.

$[sorted\_index, sorted\_distance] = sort(\mathbf{d}, 'ascend')$

Output: Pruned sample matrix $\mathbf{Y}' = \mathbf{Y}(1:sorted\_index(1:K))$.

3.4 Model construction and update

To construct the template in the first frame, we manually select the first positive template and obtain the rest $N_p - 1$ of the positive templates by perturbating one pixel in different directions. Then, the N_n negative samples are randomly sampled from the set $\Omega = \left\{ r,c \,|\, \frac{w}{8} < |r - r^*| < \frac{w}{4}; \frac{h}{8} < |c - c^*| < \frac{h}{4} \right\}$, where (r^*, c^*) denotes the optimal locations (the location of the target in the first frame or the tracked result in the rest of the frames); w and h indicate the width and height of the target, respectively.

To ensure that the tracker will adapt to the appearance changes of both the object and the background, introducing an online model updating mechanism into a given tracker is necessary. Thus, template set $\mathbf{A}$ should be updated during the tracking process. To update the positive sample set, we first find the nearest neighbor of the optimal observation sample $\hat{\mathbf{y}}$ in the positive template set $\mathbf{A}^+$ and obtain the nearest neighbor $\mathbf{a}_{\hat{j}}^+$ ($\hat{j} = \arg\min_j \left\| \hat{\mathbf{y}} - \mathbf{a}_j^+ \right\|_2^2, j = 1,2,...,N_p$). Then, we replace $\mathbf{a}_{\hat{j}}^+$ with the observation sample $\hat{\mathbf{y}}$. To update the negative sample set, we discard all the old negative samples in $\mathbf{A}^-$ and resample

N_n samples using a procedure similar to that in the first frame to build a new negative template set.

3.5 Tracking framework

On the basis of the preceding presentations and discussions, we summarize the framework of our tracker in **Algorithm 3**.

Algorithm 3. Framework of the proposed tracking algorithm.

Step 1: Initialize the parameters of our tracker manually.

Step 2: Collect positive and negative samples to build template set $\mathbf{A}_1$ in the first frame.

Step 3: Perform tracking

For $t = 2$ *to* T (T is the total frame number) **do**

(1) Sample N candidate states $\mathbf{Z}_t = \left[\mathbf{z}_t^1, \mathbf{z}_t^2, ..., \mathbf{z}_t^N\right]$ and extract their corresponding locally normalized features $\mathbf{Y}_t = \left[\mathbf{y}_t^1, \mathbf{y}_t^2, ..., \mathbf{y}_t^N\right]$ (refer to Section 3.1 for additional details).

(2) Prune candidate samples $\mathbf{Y}_t$ into a small pruned sample matrix $\mathbf{Y}_t^{'}$ (Section 3.3).

(3) Solve the structured sparse representation model $\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \left\|\mathbf{Y}_t^{'} - \mathbf{A}_t\mathbf{X}\right\|_F^2$, $\left\|\mathbf{X}\right\|_{row,0} \le T_0$ by using the SOMP method (Section 3.2)

(4) Compute the observation likelihood $p\left(\mathbf{y}_t^i \mid \mathbf{z}_t^i\right) = \exp\left[-\left(\left\|\mathbf{y}_t^i - \mathbf{A}_t^{+}\mathbf{x}_i\right\|_2^2 - \left\|\mathbf{y}_t^i - \mathbf{A}_t^{-}\mathbf{x}_i\right\|_2^2\right)\right]$.

(5) Infer the optimal state of the tracked object using the particle filter framework.

(6) Collect positive and negative samples to update the template set (Section 3.4).

End For

4. Experimental Results and Discussions

In this work, our tracker is implemented in MATLAB platform and ran at 13 frames per second on a PC with Intel i7-3770 CPU (3.4 GHz) with 32 GB memory. The variance matrix of the affine parameters is set to $\boldsymbol{\psi} = diag\left(4, 4, 0.01, 0.005, 0.001, 0.001\right)$ as the default values. For each video clip, the bounding box of the tracked object is manually labeled in the first frame. We resize each observation patch to 32×32 pixels and then extract its locally normalized feature vector using 4×4 fragments. As a trade-off between effectiveness and speed, we adopt 600 particles and select 100 of these particles using the proposed particle

selection mechanism. The numbers of positive and negative samples are set to 10 and 50, respectively. The sparsity level T_0 of the representation model is set to 7. The effects of several critical parameters are discussed in a later section.

For experimental comparisons, we collect nine challenging image sequences, with challenging factors that include partial occlusion, pose change, illumination variation, scale change, and background clutter. By adopting these challenging video clips, our tracker is compared with eight state-of-the-art tracking algorithms, including the fragment-based tracking (FragT) [6], IVT [16], MIL [12], visual tracking decomposition (VTD) [7], tracking-learning-detection (TLD) [14], accelerated proximal gradient L1 (APGL1) [21], multitask tracking (MTT) [20], and local sparse appearance tracking (LSAT) [27] algorithms.

4.1 Qualitative comparisons

Fig. 3 illustrates that the proposed tracking algorithm achieves good performance (in terms of position, rotation, and scale) when the tracked target experiences partial occlusions during the tracking process, which can be mainly attributed to three reasons. First, the structured sparse representation model is used in our appearance model to utilize the relationships between candidates and templates as well as the relationships among different candidates (i.e., different candidates share similar sparse patterns). Second, the adopted locally normalized feature causes our tracker to be less sensitive to partial occlusion. Third, the negative samples allow our tracker to be less sensitive to noises from the background. Notably, the IVT algorithm performs poorly in handling occlusion because the Gaussian noise assumption is ineffective in modeling abnormal outliers. The FragT method handles partial occlusion via fragment-based object representation with an integral histogram. Although satisfying results are achieved in a few simple examples (e.g., *Occlusion1*), this method unsatisfactorily works in more challenging cases (e.g., *Occlusion2* and *Caviar2*) because it cannot address appearance changes caused by scale and pose. The MIL method aims to solve ambiguity, but achieves unsatisfactory performance when the targets are occluded by similar objects because the binary features used are ineffective in distinguishing objects with similar appearances (e.g., *Caviar1* and *Caviar2*). Although the APGL1 tracker considers partial occlusion explicitly by using a set of trivial templates, this method also exhibits poor performance in some cases (e.g., *Caviar1* and *Caviar2*) because it does not model the relationships among candidates and does not utilize background information.

In addition, **Figs. 4** and **5** present the representative results for the other challenging image sequences. The results show that our tracker also performs efficiently in handling other challenging factors, including pose change (*DavidIndoor*), illumination variation (*Singer* and *Car1*), and background clutter (*Car2* and *Deer*).

Fig. 3. Representative results for the *FaceOcc1*, *FaceOcc2*, *Caviar1*, and *Caviar2* sequences, which highlight that the tracked objects exhibit severe occlusion.

Fig. 4. Representative results for the *DavidIndoor*, *Singer*, and *Car1* sequences, which highlight out-of-plane rotation, scale change, and illumination variation.

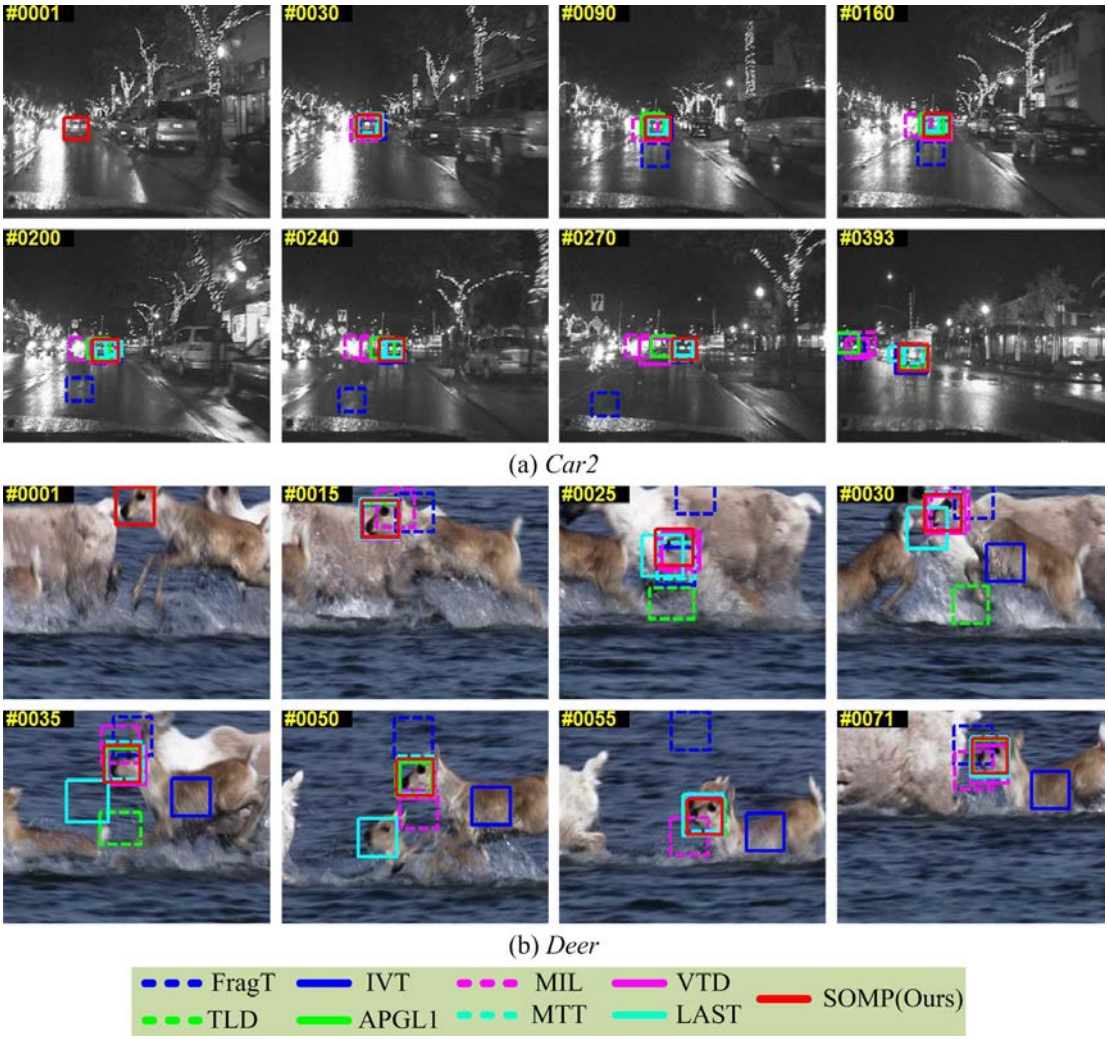

Fig. 5. Representative results for the *Car2* and *Deer* sequences, which highlight background clutter and motion blur.

4.2 Quantitative comparison

In this study, we adopt two standard criteria to evaluate the proposed tracking algorithm and other similar trackers. The first rule is center error (CE), which is defined as

$$CE(t) = \sqrt{\left[x_T(t) - x_G(t)\right]^2 + \left[y_T(t) - y_G(t)\right]^2}, \tag{12}$$

where $\{x_G(t), y_G(t)\}$ denotes the center location (i.e., horizontal and vertical coordinates) of the ground truth, $\{x_T(t), y_T(t)\}$ denotes the center location obtained by a given tracker, and t is the frame index. Evidently, a tracker that exhibits good performance intends to achieve small CE values in each sequence. **Table 1** provides the average CE (i.e., $ACE = \frac{1}{T}\sum_{t=1}^{T} CE(t)$, T is the number of total frames) values of these trackers. The red text denotes the smallest CE values, whereas the blue text denotes the second best values. Although the center location

error is highly intuitive, the scale and rotation changes of the tracked objects cannot be considered. Therefore, we also adopt the second rule, i.e., overlap rate (OR), which is defined as

$$OR(t) = area\{\mathbf{B}_T(t) \cap \mathbf{B}_G(t)\} / area\{\mathbf{B}_T(t) \cup \mathbf{B}_G(t)\}, \quad (13)$$

where $\mathbf{B}_G(t)$ and $\mathbf{B}_T(t)$ are the ground truth bounding box and the tracked bounding box in the t-th frame, respectively. **Table 2** summarizes the average OR, i.e., $AOR = \frac{1}{T}\sum_{t=1}^{T} OR(t)$, values of these trackers, where the red text denotes the best performance and the blue text denotes the second best result. As observed from **Tables 1** and **2**, our tracker performs better than the other state-of-the-art tracking algorithms in some challenging image sequences.

Table 1. Average CE of the tracking algorithms. The best three results are shown in red (first), blue (second) and green (third) text.

Method / Sequence	FragT [6]	IVT [16]	MIL [12]	VTD [7]	TLD [14]	APGL1 [21]	LSAT [20]	MTT [27]	Ours
FaceOcc1	**5.6**	9.2	32.3	11.1	17.6	6.8	**5.3**	14.1	**4.5**
FaceOcc2	15.5	10.2	14.1	10.4	18.6	**6.3**	58.6	**9.2**	**3.4**
Caviar1	5.7	45.2	48.5	**3.9**	5.6	50.1	**1.8**	20.9	**1.4**
Caviar2	**5.6**	8.6	70.3	**4.7**	8.5	63.1	45.6	65.4	**3.8**
DavidIndoor	76.7	**3.6**	16.1	13.6	9.7	14.3	**4.9**	124.0	**4.3**
Singer	22.0	8.5	15.2	**4.1**	32.7	**3.1**	14.5	41.2	**4.0**
Car1	179.8	**2.9**	60.1	12.3	18.8	16.4	**3.3**	37.2	**3.1**
Car2	63.9	**2.1**	43.5	27.1	25.1	**1.7**	4.1	**1.8**	**1.7**
Deer	92.1	127.5	66.5	**11.9**	25.7	38.4	69.8	**9.2**	**10.1**
Average	51.9	24.2	40.7	**11.0**	**18.0**	22.2	23.1	35.9	**4.0**

Table 2. Average OR of the tracking algorithms. The best three results are shown in red (first), blue (second) and green (third) text.

Method / Sequence	FragT [6]	IVT [16]	MIL [12]	VTD [7]	TLD [14]	APGL1 [21]	LAST [20]	MTT [27]	Ours
FaceOcc1	**0.90**	0.85	0.59	0.77	0.65	**0.87**	**0.90**	0.79	**0.91**
FaceOcc2	0.60	0.59	0.61	0.59	0.49	**0.70**	0.33	**0.72**	**0.84**
Caviar1	0.68	0.28	0.25	**0.83**	0.70	0.28	**0.85**	0.45	**0.88**
Caviar2	0.56	0.45	0.26	**0.67**	**0.66**	0.32	0.28	0.33	**0.76**
DavidIndoor	0.19	**0.71**	0.45	0.53	0.60	0.57	**0.62**	0.27	**0.78**
Singer	0.34	**0.66**	0.34	**0.79**	0.41	**0.83**	0.52	0.32	**0.83**
Car1	0.22	**0.92**	0.34	**0.73**	0.64	0.70	**0.91**	0.53	**0.91**
Car2	0.09	**0.81**	0.17	0.43	0.38	**0.83**	0.49	0.58	**0.80**
Deer	0.08	0.22	0.21	**0.58**	0.41	0.45	0.35	**0.60**	**0.63**
Average	0.41	0.61	0.36	**0.66**	0.55	**0.62**	0.58	0.51	**0.80**

4.3 Effects of critical parameters

In this subsection, we investigate the effects of several critical parameters on our tracker and present the results in **Tables 3**, **4**, and **5**, in which ACE denotes average CE, AOR denotes average OR, and FPS indicates frame per second, which measures the speed of a given tracker.

First, we report the tracking results with different sparsity levels in **Table 3**. If the sparsity level is too small, then a sufficient number of templates to represent candidates cannot be selected although tracking speed is extremely high. However, if the sparsity level is too large, then several noise templates may be selected to represent all the candidates. Thus, the sparsity level should be set to an appropriate value ($T_0 = 7$ is used in this work). Second, the number of fragments should also be an appropriate value (the corresponding experimental results are provided in **Table 4**). If the number is too small, then the tracker cannot address abnormal noises, which may lead to tracking drift (such as $M = 1$). By contrast, a large number of fragments causes object representation to be highly trivial, which results in the poor accuracy and fast speed of the tracker (e.g., $M = 6$). Finally, the effects of different selection numbers on selecting candidates are presented in **Table 5**. The result shows that the proposed discriminative candidate selection method is highly effective in terms of both accuracy and speed.

Table 3. Effect of the sparisity level for visual tracking

Measure \ T_0	1	3	5	**7**	9	11
ACE	36.0	17.4	8.4	**4.0**	5.9	16.6
AOR	0.62	0.70	0.75	**0.82**	0.80	0.73
FPS	16	15	14	13	12	12

Table 4. Effect of the number of fragments

Measure \ M	1	2	**4**	6	8
ACE	25.4	11.7	**4.0**	24.6	10.6
AOR	0.62	0.75	**0.82**	0.71	0.79
FPS	16	14	13	11	8

Table 5. Effect of the selected number for visual tracking

Measure \ K	50	100	200	400	600
ACE	5.5	**4.0**	15.7	34.8	67.0
AOR	0.79	**0.82**	0.77	0.61	0.46
FPS	15	13	11	8	7

5. Conclusion

In this study, we present a novel object-tracking method based on the proposed structured sparse representation model. First, we simultaneously represent all the candidate samples using a series of object and background templates, which can be solved via the SOMP method. Second, we develop a tracking algorithm based on the particle filter framework by comparing our structured sparse representation model with a discriminative candidate selection scheme and a simple updating method. Finally, numerous experiments are conducted to evaluate the proposed tracker and to compare this tracker with other state-of-the-art algorithms. Both qualitative and quantitative experimental results demonstrate that the proposed tracker performs better than the other algorithms.

References

[1] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony R. Dick and Anton van den Hengel, "A survey of appearance models in visual object tracking," A*CM Transactions on Intelligent Systemsand Technology s*, vol. 4, no. 4, pp. 58, December, 2013. Article (CrossRef Link)

[2] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang and Zhan Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol.74, no.18, pp.3823-3831, November, 2011. Article (CrossRef Link)

[3] Dorin Comaniciu, Visvanathan Ramesh and Peter Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.5, pp.564-575, May, 2003. Article (CrossRef Link)

[4] P. P´erez, C. Hue, J. Vermaak and M. Gangnet, "Color-based probabilistic tracking," in *Proc. of European Conference on Computer Vision*, pp.661-675, May 28–31, 2002. Article (CrossRef Link)

[5] Yuan Li, Haizhou Ai, Takayoshi Yamashita, Shihong Lao and Masato Kawade, "Tracking in Low Frame Rate Video: A cascade particle filter with discriminative observers of different life spans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1728–1740, October, 2008. Article (CrossRef Link)

[6] Amit Adam, Ehud Rivlin and Ilan Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 798–805, June 17-22, 2006. Article (CrossRef Link)

[7] Junseok Kwon and Kyoung Mu Lee, "Visual tracking decomposition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, June 13-18, 2010. Article (CrossRef Link)

[8] Shengfeng He, Qingxiong Yang, Lau, R.W.H., Jiang Wang and Ming-Hsuan Yang, "Visual Tracking via Locality Sensitive Histograms", in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2427-2434, June 23-28, 2013. Article (CrossRef Link)

[9] Shai Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp.261-271, January, 2007. Article (CrossRef Link)

[10] Shai Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp.1064-1072, September, 2004. Article (CrossRef Link)

[11] Helmut Grabner and Horst Bischof, "On-line boosting and vision," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 260-267, June 17-22, 2006. Article (CrossRef Link)

[12] Boris Babenko, Ming-Hsuan Yang and Serge Belongie, "Robust Object Tracking with Online Multiple Instance Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.33, no.8, pp.1619-1632, August, 2011. Article (CrossRef Link)

[13] Fan Yang, Huchuan Lu and Minghsuan Yang, "Robust Superpixel Tracking," *IEEE Transaction on Image Processing*, vol.23, no.4, pp.1639-1651, April, 2014. Article (CrossRef Link)

[14] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, July, 2012. Article (CrossRef Link)

[15] Sam Hare and Amir Saffari and Philip H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. of International Conference on Computer Vision*, pp. 263-270, November 6-13, 2011. Article (CrossRef Link)

[16] David A. Ross, Jongwoo Lim, Ruei-Sung Lin and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, August, 2008. Article (CrossRef Link)

[17] Xi Li, Weiming Hu, Zhongfei Zhang, Xiaoqin Zhang and Guan Luo, "Visual tracking via incremental log-Euclidean Riemannian subspace learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, June 23-28, 2008. Article (CrossRef Link)

[18] Xue Mei and Haibin Ling, "Robust visual tracking using L1 minimization," in *Proc. of International Conference on Computer Vision*, pp. 1436-1443, September 29-October 2, 2009. Article (CrossRef Link)
[19] Dong Wang, Huchuan Lu and Minghsuan Yang, "Online object tracking with sparse prototypes," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 314-325, January, 2013. Article (CrossRef Link)
[20] Tianzhu Zhang, Bernard Ghanem, Si Liu and Narendra Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2042-2049, June 16-21, 2012. Article (CrossRef Link)
[21] Chenglong Bao, Yi Wu, Haibin Ling and Hui Ji , "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830-1837, June 16-21, 2012. Article (CrossRef Link)
[22] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S. Huang and Shuicheng Yan, "Sparse Representation for Computer Vision and Pattern Recognition," in *Proc of the IEEE*, vol.98, no.6, pp.1031-1044, June, 2010. Article (CrossRef Link)
[23] John Wright, Allen Y. Yang, Arvind Ganesh, S.Shankar Sastry and Yi Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, February, 2009. Article (CrossRef Link)
[24] Xue Mei, Haibin Ling, Yi Wu, Erik Blasch and Li Bai, "Minimum error bounded efficient L1 tracker with occlusion detection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1257-1264, June 20-25, 2011. Article (CrossRef Link)
[25] Kaihua Zhang, Lei Zhang and Ming-Hsuan Yang, "Fast Compressive Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no.10, pp. 2002-2015, September, 2014. Article (CrossRef Link)
[26] Yi Chen, Nasser M. Nasrabadi and Trac D. Tran, "Hyperspectral Image Classification Using Dictionary-Based Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no.10, pp. 3973-3985, October, 2011. Article (CrossRef Link)
[27] Baiyang Liu, Junzhou Huang, Lin Yang and Casimir Kulikowsk, "Robust tracking using local sparse appearance model and K-selection," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1313-1320, June 20-25, 2011. Article (CrossRef Link)

Chunjuan Bo received the B.E. degree in Electronic Information Engineering, Dalian Nationalities University, China, in 2008. She also received M.S. degree in Communication and Information System, Dalian University of Technology, China, in 2010. She is currently a faculty in the College of Electromechanical Engineering of Dalian Nationalities University. Her research interests include image classification, pattern recognition and so on.

Dong Wang received the B.E. degree in electronic information engineering and the Ph.D. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 2013, respectively. He is currently a Faculty Member with the School of Information and Communication Engineering, DUT. His current research interests include face recognition, interactive image segmentation, and object tracking.