J. Vis. Commun. Image R. 25 (2014) 423–434

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Collaborative object tracking model with local sparse representation

Chengjun Xie [a,b], Jieqing Tan [a], Peng Chen [c,*], Jie Zhang [b], Lei He [a]

[a] School of Computer & Information, Hefei University of Technology, Hefei 230009, China
[b] Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China
[c] Institute of Health Sciences, Anhui University, Hefei, Anhui 230601, China

ARTICLE INFO

Article history:
Received 16 August 2013
Accepted 7 December 2013
Available online 16 December 2013

Keywords:
Object tracking
Discriminative model
Generative model
Sparse representation
Appearance model
Collaborative model
Sparse coding histogram
Similarity measure

ABSTRACT

There existed many visual tracking methods that are based on sparse representation model, most of them were either generative or discriminative, which made object tracking more difficult when objects have undergone large pose change, illumination variation or partial occlusion. To address this issue, in this paper we propose a collaborative object tracking model with local sparse representation. The key idea of our method is to develop a local sparse representation-based discriminative model (SRDM) and a local sparse representation-based generative model (SRGM). In the SRDM module, the appearance of a target is modeled by local sparse codes that can be formed as training data for a linear classifier to discriminate the target from the background. In the SRGM module, the appearance of the target is represented by sparse coding histogram and a sparse coding-based similarity measure is applied to compute the distance between histograms of a target candidate and the target template. Finally, a collaborative similarity measure is proposed for measuring the difference of the two models, and then the corresponding likelihood of the target candidates is input into a particle filter framework to estimate the target state sequentially over time in visual tracking. Experiments on some publicly available benchmarks of video sequences showed that our proposed tracker is robust and effective.

1. Introduction

Object tracking is one of the most important components in computer vision and arises in many practical applications such as video surveillance, human motion understanding, and interactive video processing, and so on. Although many trackers have been proposed and have made successes under various scenarios, object tracking is still challenging because the appearance of an object may be changed drastically while undergoing significant pose change, illumination variation and/or partial occlusion. Such a thorough review can be found in [1,12], where tracking algorithms were categorized into generative and discriminative approaches. Generative methods formulated the tracking problem as searching for the most similar regions to the target model. Discriminative methods treated the tracking problem as a binary classification problem which attempts to design a classifier to distinguish the target object from the background. In this paper, we concentrate mainly on designing a robust tracking model that confronts the aforementioned challenges by combining tracking outputs of the generative and discriminative models.

Recently, sparse representation [4] has been successfully applied in visual tracking, and a plethora of sparse representation-based tracking methods have been proposed [2,5–10,19,21,23]. Among these generative appearance models based on sparse representation, tracking problems were formulated to attempt to jointly estimate the target appearance by finding a sparse linear combination over a dictionary containing the target and trivial templates. Further experiments showed that sparse representation was efficient and adaptable to the aforementioned challenges, especially to partial occlusion. However, those sparse representation-based trackers only considered global templates, did not make full use of local representations, and hence failed in tracking target when the templates directly cropped from target image are very limited [24]. Therefore, Local patch-based sparse representation models were introduced in [5,7,9]. In [5,7] the object appearance was modeled by histograms of local sparse representation, however, both of their methods were based on a static local dictionary obtained from the first frame and may fail in dynamic scenes. Afterwards, Jia et al. [9] adopted an alignment pooling method scanning across local patches based on sparse coefficients for robust tracking. Although these trackers with local sparse representation have demonstrated good robustness in many videos, they may fail in the discrimination between the target and the background more possible when there are some challenging factors, such as background clutter and the background regions with similar

* Corresponding author.
E-mail addresses: cjxie@iim.ac.cn (C. Xie), bigeagle@mail.ustc.edu.cn (P. Chen).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.12.012

appearance to the object class. In comparison, discriminative appearance models based on sparse representation posed visual object tracking as a binary classification issue. In [2], local image patches of a target object were represented by their sparse codes with an over-complete dictionary constructed online, and the sparse codes were treated as training samples. The key idea of the model was to train a classifier by learning the sparse codes and to maximize the separability between the object and non-object regions discriminately. Nevertheless, a major limitation of the discriminative appearance models is in that they were heavily relied on training sample selection. Moreover, most discriminative appearance models took the current object location as one positive sample, and its neighborhoods as negatives. However, the imprecise current object location could degrade the appearance model and cause drift.

In actually, generative and discriminative appearance models have their respective pros and cons, and are complementary to each other to a certain extent. Therefore, we propose an efficient tracking algorithm incorporating the information from a developed generative model and a discriminative model, based on local sparse representation. Our proposed algorithm mainly contain a local dictionary obtained by sampling local image patches within the target region from the first frame; a sparse representation-based generative model (SRGM) with local sparse template which is represented by histograms of local sparse representation based on the dictionary and is updated online; a sparse representation-based discriminative model (SRDM) with a linear classifier which is trained by learning the sparse codes based on the dictionary and is updated online; and a similarity function fusing the information from the generative model and the discriminative appearance model. The discriminative model is able to investigate informative samples as support vectors for object/non-object classification, resulting in a strong discrimination. Although our SRDM module has a good generalization ability to distinguish object and background, the local sparse representation-based classifier may be affected when updated with the background information as positive samples. However, the SRGM module can alleviate the influence since it is distinct to be foreground or background with local sparse coding histograms. Thus, the SRDM and SRGM module are complementary to each other to some extent.

The main contributions of this paper are:

- A novel target appearance modeling method by combining the generative model and the discriminative appearance model based on local sparse representation.
- A new similarity measure between the candidates by fusing the information from the generative model and the discriminative appearance model.

2. Related works

There was a rich literature on appearance modeling and representation [12]. An effective object representation should have a strong description or discrimination ability to distinguish targets from background. Most of recent tracking algorithms focused on object representation schemes with generative appearance models [3,6–10,13,14,16,17,19–23,33,38] and discriminative models [2,5,11,15,25,26,37].

Generative methods formulated the tracking problem as searching for the most similar regions to the target model. Intensity histogram was perhaps the simplest way to represent object appearance in many tracking algorithms [3], but it missed the spatial information of object appearance, which makes it sensitive to noise as well as occlusion in many tracking applications. To solve these problems, Nejhum et al. [17] modeled the target appearance as a small number of rectangular blocks with histograms, whose positions within the tracking window are determined adaptively. More recently, He et al. [22] presented a tracking framework based on a locality sensitive histogram that was computed at each pixel location and a floating-point value was added to the corresponding bin for each occurrence of an intensity value. In addition, to cover a wide range of pose and illumination variation, Ross et al. developed an online subspace learning model to account for appearance variation [13]. Recently, the sparse representation framework [4,18] has attracted considerable interests in object tracking due to its robustness to occlusion and image noise. Following the pioneer work, many methods adopted sparse representation model for tracking objects [5–10,19–21,23]. In [6], each target candidate was represented as a linear combination of a set of online updated templates, consisting of target templates and trivial templates, and the candidate with the smallest error to target template reconstruction is regarded as the tracking result. More recently, Zhang et al. [8] presented a multi-task sparse optimization framework. Instead of treating test samples independently, the framework explored the interdependencies between test samples by solving a regularized group sparsity problem. Besides the high computational cost, another drawback of these trackers is to model object appearance as global sparse templates. Since local representations can capture the local structural target appearance, the local visual representations [7,9,22] were robust to global appearance changes caused by illumination variation, shape deformation, and partial occlusion. In [24], extensive experiments have demonstrated that local sparse representation-based trackers outperformed those with global sparse templates. Therefore, Jia et al. [9] adopted an alignment pooling method scanning across local patches based on sparse coefficients for robust tracking. As these methods exploited generative representation of target objects only and did not take the background into account, they were less effective for tracking in cluttered background.

By training a model via a discriminative classifier, discriminative methods [2,11,15,25,26,37] have shown good performance in discriminating object from the background. Avidan et al. [11] developed an online boosting method for tracking targets, which was an ensemble tracker that yielded a strong classifier by a set of weak classifiers. Bai et al. [25] treated object tracking as a weakly supervised ranking problem, which can avoid the heuristic and unreliable step of training sample selection towards the true target samples. In contrast with them, Babenko et al. [15] used multiple instance learning (MIL) instead of traditional supervised learning to handle ambiguous binary data obtained online. Zhang et al. [26] proposed an online weighted multiple instance tracker, which incorporated the important information of samples into the online multi-instance boosting learning process, resulting in robust tracking results. Despite the success of the discriminative methods, a major challenge is how to choose positive and negative samples when updating the adaptive appearance model. Since most discriminative trackers took the current object location as one positive sample and sampled its neighbors as negatives, it might degrade the appearance model and cause drift due to the imprecise current object location.

In this paper, we propose an effective object tacking method involving a generative model and a discriminative appearance model based on local sparse representation. The proposed method consists of three main parts: a generative appearance model for object representation, which is composed of local patch templates with the corresponding histograms of local sparse representation, and thus provides a more flexible mechanism to deal with the problem of appearance change; a discriminative appearance model for object representation, which is obtained by learning the local sparse codes of the negative and positive samples, and thus is capable of discriminating object from the background powerfully; a similarity mea-

sure, which is the combination of different models between the generative model and the discriminative appearance model, and further ensures a more stable scheme to locate the target more accurately.

3. Proposed algorithm

While most tracking algorithms use either generative or discriminative appearance model, our module collaborates both of them. In general, discriminative methods use information from the target and the background, yet the discriminative methods may be affected when updated with the background patches as positive samples. On the contrary, generative methods are often distinct to be target or background by searching for the most similar regions to the target in the current frame. Thus, generative methods are more amenable for discriminative models because of their simplicity and flexibility.

The basic flow of our collaborative appearance model is illustrated in Fig. 1. The main procedures of our model are: for SRDM module, firstly positive and negative samples around the manually labeled target location are cropped out, then sparse codes for each image patch are computed to form the training data and thus, a linear classifier is constructed by learning the sparse codes; for the SRGM module, overlapped sliding windows on the normalized target image is applied to obtain image patches, then each image patch using an over-complete dictionary is encoded, and the corresponding sparse codes are aggregated to form sparse coding histogram; finally, our target template is generated by the sparse coding histogram.

3.1. Object representation by local sparse coding

Motivated by the success of sparse representation in object tracking [5,7], a local sparse representation is used to model the appearance of target patches and the corresponding sparse codes are collected to represent the target object. Let $P = \{p_i | i = 1{:}K\}$ denote the vectorized image patches extracted from a target region, where $p_i \in R^d$ ψ is the ith local image patch, d is the dimensionality of image patches, and K is the number of local image patches. Each local image patch p_i from the target region will have a corresponding vector with sparse coefficient $\alpha_i \in R^{c \times 1}$, which is computed by:

$$\hat{\alpha}_i = \arg\min \|\alpha_i\|_1 \text{ subject to } \|p_i - D\alpha_i\|_2 < \varepsilon, \tag{1}$$

where the dictionary $D \in R^{d \times c}$ is generated from k-means cluster centers (c denotes the number of clusters) via the patches belonging to the labeled target object in the first frame. When the sparse codes $A_j = [\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_K]^{\mathrm{T}}$ of all the image patches for the jth candidate are computed, they are collected to represent the whole candidates.

3.2. Sparse representation-based discriminative model

To initialize the classifier of the model in the first frame, positive and negative samples (local image patches) are cropped out around the labeled target location. Let $l(x)$ denote the location of image patch x. First, a set of image patches $X^r = \{x | r > \|l(x) - l_1^*\|\}$ for positive samples is cropped out from the first frame, and image patches from an annular region $X^{r,\beta} = \{x | r < \|l(x) - l_1^*\| < \beta\}$ are regarded as negative samples, where r and β are thresholds defining the annular area (measured in pixels), respectively. Then the sparse codes of image patches are computed to form the training data, $\{(A_1, y_1), \ldots, (A_n, y_n)\}$, where $A_j \in R^{(K \times c) \times 1}$, $y_j \in \{+1, -1\}$, and n is the number of training samples.

With the training data, our discriminative model is learned by minimizing the following loss function

$$\min_{w,b} L(w, b) = \min_{w,b} \sum_{j=1}^{n} l(y_j, w, b, A_j) + \frac{\lambda}{2}\|w\|^2, \tag{2}$$

where $l(y_j, w, b, A_j)$ is a loss function, w and b are classifier parameters. Apart from convexity, smoothness is another desirable fact for the loss function, so the exponential loss function has the form as below:

$$l(y, w, b, A) = e^{-y(w^{\mathrm{T}}A + b)}. \tag{3}$$

In the tracking process, candidates are sampled and the corresponding sparse codes are calculated as the form $A = \{A_1, \ldots, A_N\}$, where N is the number of candidates. Then the classification score for the learned classifier can be obtained by

$$score(A) = e^{w^{\mathrm{T}}A + b}. \tag{4}$$

The basic flow of our SRDM appearance model is illustrated in Fig. 2.

3.3. Sparse representation-based generative model

To represent the basis distribution for both target and candidates, similar to [7], the sparse coding histogram is defined by:

$$T_q = C\sum_{i}^{K} k(\|c_i\|^2)|a_{iq}|, \tag{5}$$

where T_q is the value of the qth bin in sparse coding histogram, c_i represents local patch i at a certain position of the target, $k(\|c_i\|^2)$ is an isotropic kernel function which is applied to assign smaller weights to image patch far away from the target center, C is a normalization constant, and a_{iq} is the qth coefficient of the ith image patch. Similarly, the sparse coding histogram of a candidate can be computed as:

$$H_q = C\sum_{i}^{K} k\left(\left\|\frac{Y - c_i}{h}\right\|^2\right)|a_{iq}^*|, \tag{6}$$

where Y represents certain position of the target.

How to determine the similarity between the candidate and the template is one key issue in object tracking so that the most similar target location can be found out by a similarity measure function. Therefore, let $sim(T, H)$ be a similarity measure between the candi-

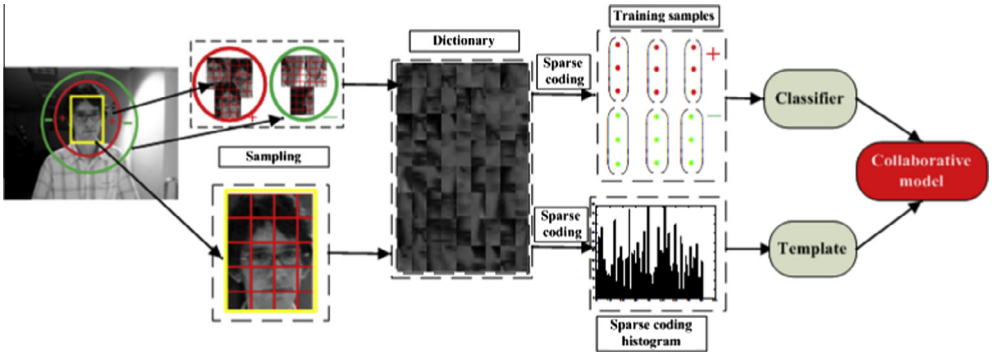


Fig. 1. The basic flow of our collaborative appearance model. It consists of two main parts: SRDM module and SRGM module.

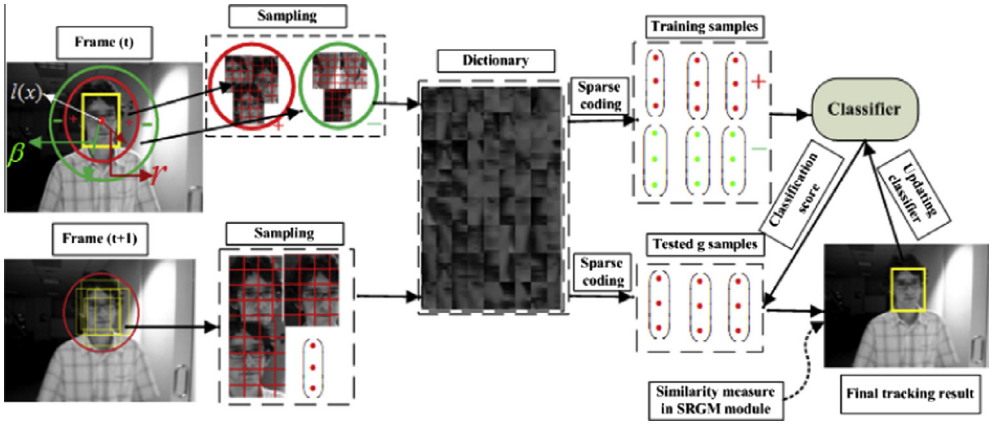


Fig. 2. Graphical representation of SRDM module.

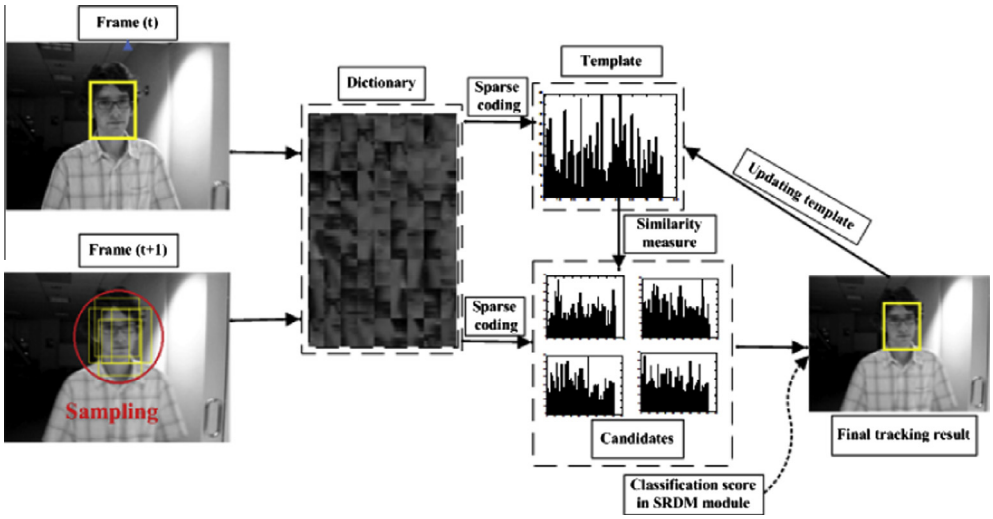


Fig. 3. Graphical representation of SRGM module.

date H and the template T to search for the most similar candidate. There are some well-known goodness-of-fit statistical metrics such as Euclidean distance, Bhattacharyya coefficient and log-likelihood ratio statistic [35]. Here the weighted sum of Bhattacharyya distance is adopted to measure the similarity as

$$sim(T,H) = \rho(T,H), \tag{7}$$

where ρ is the Bhattacharyya distance between the histograms of candidate and template,

$$\rho(T,H) = \sum_{q=1}^{c} \sqrt{T_q \cdot H_q}. \tag{8}$$

The Bhattacharyya distance can reflect the perceptual similarity between candidate and template. If they are partly similar, their histograms will be similar much more and hence the corresponding Bhattacharyya distance will be high, i.e., the angle between the two histogram vectors is small, and vice verse. Occasionally, it is possible that the histograms of two dissimilar templates are very similar. Fortunately, such cases are rare because the sparse coding histograms are local and they only reflect the local features of images [36]. The basic flow of SRGM model is shown in Fig. 3.

3.4. Collaborative object tracking model

We propose a collaborative model to integrate the discriminative module and the generative module based on local sparse representation. Here, a more robust and effective probability function for candidates is proposed, by fusing the classification score based on the learning model and the similarity function based on the sparse coding histograms. The collaborative likelihood measure of the cth candidate is defined as

$$L^c = score(A^c) * sim(T,H^c), \tag{9}$$

and the candidate with the highest probability is formed as the tracking result.

In this paper, we select the multiplicative formula, which is more effective in our tracking method than the alternative additive strategy (e.g., $L^c = \varepsilon * score(A^c) + (1-\varepsilon) * sim(T,H^c)$), to measure the likelihood of candidate. It is because that, firstly, the SRDM can learn a margin-based discriminative SVM classifier for maximizing inner-class separability. With the power of max-margin learning, the classification score $score(A^c)$ assigns larger weight to the candidate which is considered as positive samples and restricts the others. As a result, the $score(A^c)$ can be treated as the weight of the similarity function (i.e. SRGM) between the candidate and the template based on the sparse coding histograms. Moreover, in SRGM the similarity with local sparse coding histograms has the ability to reject background false alarm. The similarity $sim(T,H^c)$ assigns higher weight to the candidate with less background and simultaneously, it also be treated as the weight of the classification function (i.e. SRDM). Lastly, if the classification score of an indistinguishable candidate or the similarity of candidate with false background is approxi-

Table 1
Tracking sequences used in our experiments.

Video clip	Number of frames	Main challenges
woman	550	Partial occlusion, pose variations
shop2cor	350	Heavy occlusion, pose variations
faceocc2	814	Significant and long duration occlusion
david	462	Large illumination and pose variations, partial occlusion
bird2	98	In-plane/out-of-plane pose change and partial occlusion
sylv	1344	Illumination and pose variations
singer1	351	Significant illumination and scale variations
bolt	190	Large pose variation and fast motion

mately equal to 1, the likelihood function results in trivial change when multiplying with the classification function or the similarity function. Therefore, our model is flexible, robust and complementary to others with the simple multiplicative scheme.

3.5. Update scheme

Since object tracking with fixed template or classifier cannot adapt the change of target appearance over time, it will be failed in dynamic scenes. It is important to update the model online to enhance the adaptivity of the tracker. However, if the template or classifier is updated too frequently with new tracking results, even small errors will be accumulated and the tracker will drift from the target more and more far away. Therefore, to address the issue, firstly, the dictionary D is fixed for the same sequence so that the dictionary cannot be deteriorated even if the tracking failures and occlusions during the update process. Then, for the SRGM module, to balance between the old and new templates, a weight ω is assigned to them and the template histogram is updated by

$$H_{new} = \omega H_{first} + (1-\omega)H_{temp}, \tag{10}$$

where ω is a constant and is set to 0.8 in this paper. The new histogram H_{new} consists of the histogram H_{first} at the first frame and the histogram H_{temp} obtained during tracking process.

For SRDM module, once the tracker location is updated, the appearance model is updated every several frames. First, a set of patches $X^r = \{x | r > \|l(x) - l_1^*\|\}$ is cropped out and the positive sample is labeled. To obtain negative samples, patches $X^{r,\beta} = \{x | r < \|l(x) - l_1^*\| < \beta\}$ are cropped out from an annular region, where r and β are the same as before. Therefore the discriminative model is updated to minimize loss function of this data Eq. (2), when received new data $\{(A_1, y_1), \ldots, (A_n, y_n)\}$. In this way, our SRDM module is adaptive and discriminative.

4. Tracking by Bayesian inference

Particle filter [27,28] provided a convenient framework for estimating and propagating the posterior probability density functions of state variables. In this paper, to form a robust tracking algorithm, a collaborative similarity is embedded into the particle filter framework. Given the observations of the target $y_{1:t} = \{y_1, \ldots, y_t\}$ up to time t, the current target state S_t can be estimated by maximizing a posterior (MAP) that associates with the highest likelihood:

$$s_t = \arg\max_{S_t} p(S_t | y_{1:t}), \tag{11}$$

where $p(S_t|y_{1:t})$ is posterior probability and is recursively computed as

$$p(S_t|y_{1:t}) \propto p(y_t|S_t) \int_{S_{t-1}} p(S_t|S_{t-1}) p(S_{t-1}|y_{1:t-1}) dS_{t-1}, \tag{12}$$

where $p(y_t|S_t)$ is the observation model or likelihood function that estimates the likelihood of the state y_t given the observation S_t, while $p(S_t|S_{t-1})$ is the motion model that predicts the current state given the previous state.

4.1. Dynamic model

In this paper, similarly to [6], an affine image warping is applied to model target motion of two consecutive frames. Let $S_t = (l_1, l_2, \mu_1, \mu_2, \mu_3, \mu_4)$ be the six-dimensional parameter vector for affine transformation, where $\mu_1, \mu_2, \mu_3, \mu_4$ are the deformation parameters which represent the rotation angle, scale, aspect ratio, and skew direction at time t, respectively, and l_1, l_2 are the 2D position parameters. The transformation of each parameter is independently represented by a scalar Gaussian distribution around their previous state S_{t-1}. Then the motion model is obtained by a Gaussian distribution as follows:

$$p(S_t|S_{t-1}) = O(S_t; S_{t-1}, N), \tag{13}$$

where $O(\cdot)$ is the Gaussian distribution and N is the covariance matrix.

4.2. Observation model

In this inference framework, the observation model is very important because it reflects target appearance variations under the condition of pose changes, illumination variations or partial occlusion. Therefore, the observation model $p(y_t|S_t)$ (we omit t without causing confusion) can be defined as:

$$p(y|S) \propto L^y, \tag{14}$$

where the right side of the equation denotes the collaborative likelihood between the candidate and the target (defined in Eq. (9)) based on the local sparse coding.

Algorithm 1 gives a summary of the complete tracking algorithm.

The proposed tracking algorithm is summarized in Algorithm 1.

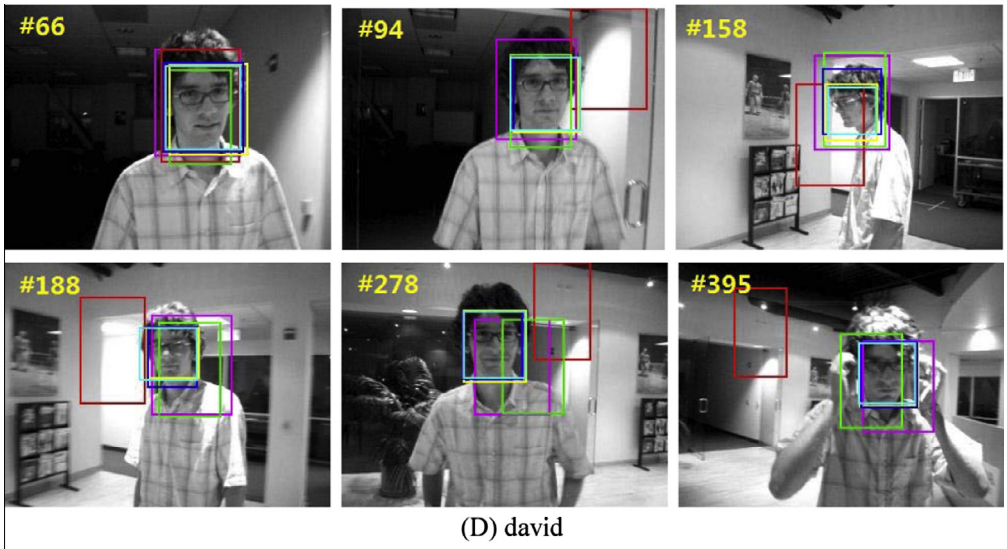
1: **Input**: The initial state of the target $S_1 = (l_1, l_2, \mu_1, \mu_2, \mu_3, \mu_4)$, the video frames $F_1, \ldots F_t$, and the dictionary D from the first frame.
2: **Output:** The current target state S_t at time t
Initialization:
3: Initialize the object in the first frame and crop out a set of negative and positive image patches.
4: Construct an initial dictionary D and compute the sparse codes of each image patch by Eq. (1).
5: Train an initial classifier by Eq. (2) and obtain classifier parameters w_1^*, b_1^*. Compute the initial template histogram by Eq. (5).
Online tracking:
6: for $t = 2, \ldots, T$ do
7: Sample candidates and calculate the corresponding sparse codes with the dictionary D by Eq. (1).
8: **SRGM module:** calculate the sparse coding histogram of the candidates and the similarity between the candidates and the template using Eqs. (6), (7) and (8).
9: **SRDM module:** compute the classification score with the learned classifier w_{t-1}^*, b_{t-1}^* by Eq. (4).
10: **Collaborative model:** determine the final tracking result S_t using the collaborative likelihood within particle filtering framework by Eqs. (11)–(14).
11: End for.
12: Update the template histogram by Eq. (10) with the newly obtained template and the classifier parameters w_t^*, b_t^*.
13: End.

Fig. 4. Screenshots of tracking comparison of our tracker (yellow box) with the trackers of L1 (red box), IVT (mulberry box), CT (green box), SCM (blue box) and ASLA (cyan box), highlighting instances of partial or significant occlusion, significant pose and illumination changes, and so on. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Experiments

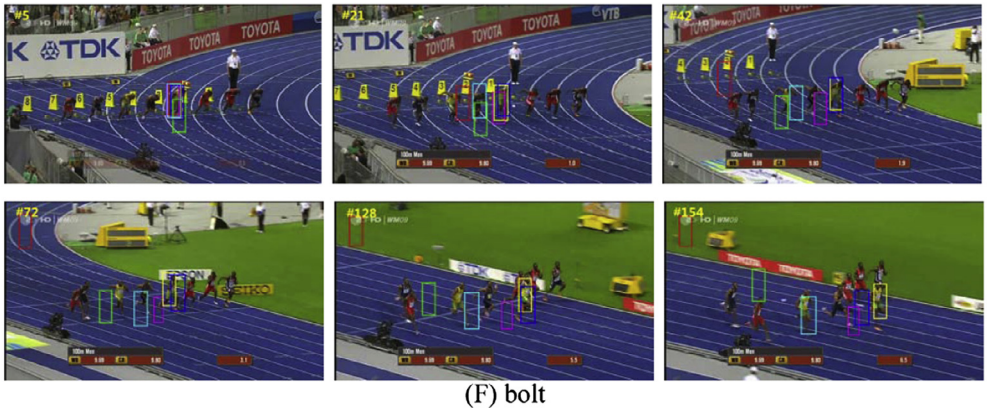
In this section, we evaluate our proposed method on eight publicly available video sequences involving the challenges of partial or significant occlusion, significant pose and illumination changes, and so on. The details of the selected video sequences are listed in Table 1. Also, five state-of-the-art trackers are tested on the same sequences, including incremental visual tracking (IVT tracker) [13], L1 tracking (L1 tracker) [6], adaptive structural local sparse appearance model (ASLA tracker) [9], sparsity-based collaborative model (SCM tracker) [5], and real-time compressive tracking (CT tracker) [29]. For fair comparison, all of them are experimented on the same dynamic model and the same particles (500 particles per frame in this work), and they use the same initialized target locations in these video sequences. The tracking videos [13,15,37], MATLAB codes, and data sets can be, respectively found from URLs [30–32].

Our method learns sparse codes and a linear classifier directly from original image patches. Therefore, the computational cost of our SRDM is smaller than that of most recent discriminative tracking methods which use multiple features. In addition, we address the computational complexity reduction of the similarity measure between candidates and template by exploiting the sparse coding histograms from original image patches. Consequently, the



(D) david

(E) bird2



(F) bolt

Fig. 4 (*continued*)

Fig. 4 (*continued*)

computational cost of our SRGM is smaller than other methods using color histograms or texture histograms. For fair comparison, all the experiments are implemented by a MATLAB on a 2.5 GHz machine with 4 GB RAM. Our method runs at around 1.5 s per frame and the IVT, L1, ASLA, SCM, CT trackers spend about 0.5, 20, 1, 1.2, 0.8 s per frame, respectively. For qualitative analysis, some representative frames are selected to show the evaluation comparison of our tracker with the others. The performance evaluation can be found in Fig. 4.

5.1. Qualitative analysis

The sequence "woman" comprises partial occlusion and pose changes. As seen in Fig. 4(A), our tracker, the trackers of ASLA and SCM show competitive compared to the trackers of L1, IVT and CT, for the whole sequence frames. On the contrary, the trackers of L1 and CT cannot adapt to these changes, resulting in serious drift (see all the frames shown in Fig. 4(A)), while the IVT tracker also fails in capturing target after frame 230.

The sequence "shop2cor" comprises heavy occlusion and pose changes. As illustrated in Fig. 4(B), our tracker performs the same with the trackers of ASLA and SCM and can track the man successfully for the whole sequence, while the trackers of IVT and CT works poorly (see the frame 106, 124, 164, 313 of shop2cor in Fig. 4(B)). The L1 tracker also performs poorly when the occlusion is severe in shop2cor (see frame 106 in shop2cor).

The sequence "faceocc2" mainly comprises heavy occlusion and pose changes. Fig. 4(C) shows that most of trackers can track the target successfully for the whole sequence, but the CT tracker appears to target drifting after frame 707.

Results on the sequence "david" are shown in Fig. 4(D). From the Fig. 4(D), the L1 tracker drifts away after the frame 94 of david sequence. The trackers of IVT and CT appear to start target drifting in frames 188, 278, 395.The other three trackers, including ours, can yield more stable and accurate results when the object undergoes the challenges of illumination changes and pose variations.

In Fig. 4(E), the trackers of IVT and CT fail to track the target after frame 35 and, the L1 as well as ASLA also miss the target in frame 52. Surprisingly, both our tracker and the SCM yield satisfactory results.

The sequence "bolt" mainly comprises very fast motion and large pose variations. From the Fig. 4(F), as shown in frames 21,

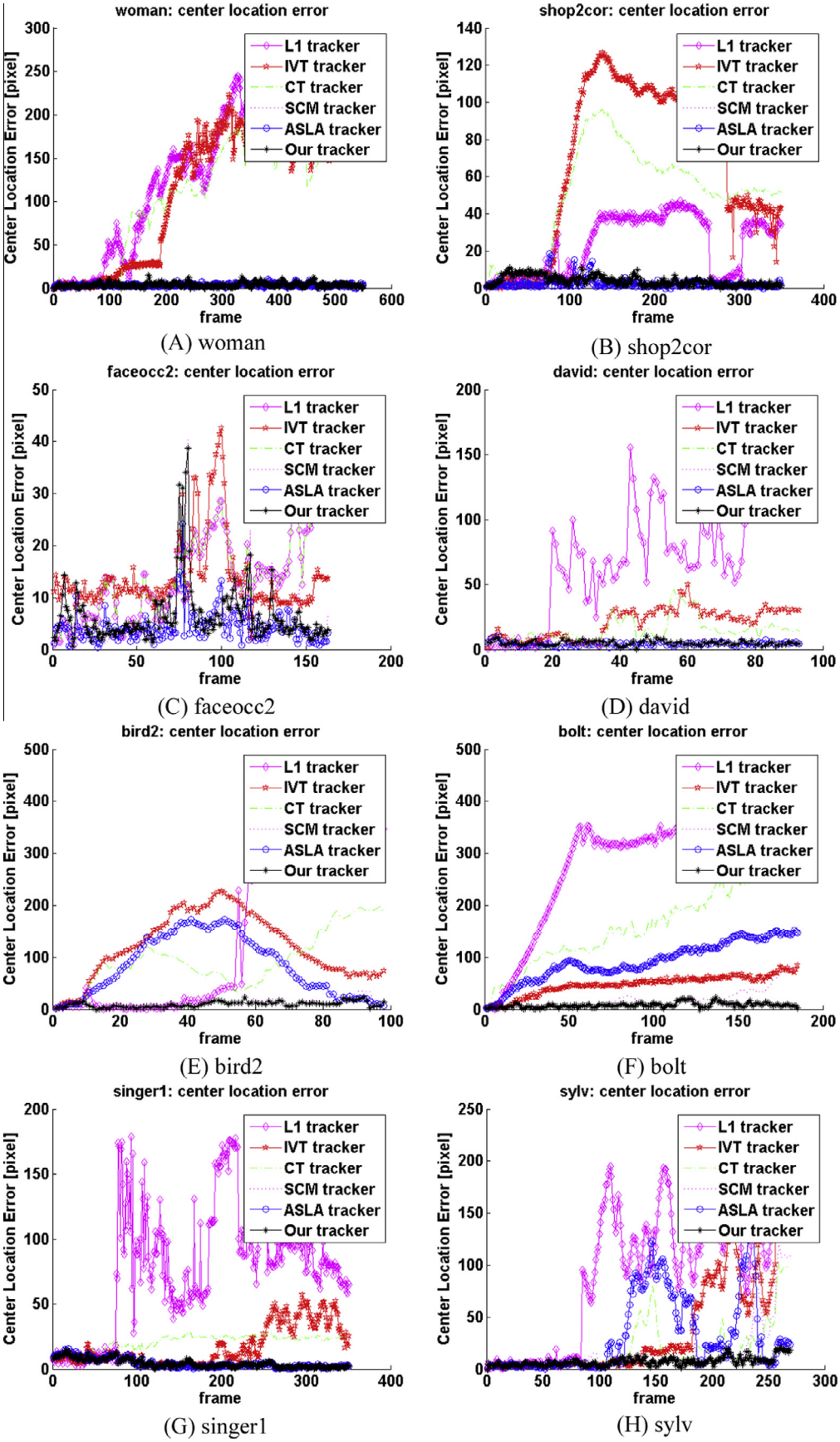


Fig. 5. Center location error plots for our tracker and the five state-of-the-art trackers.

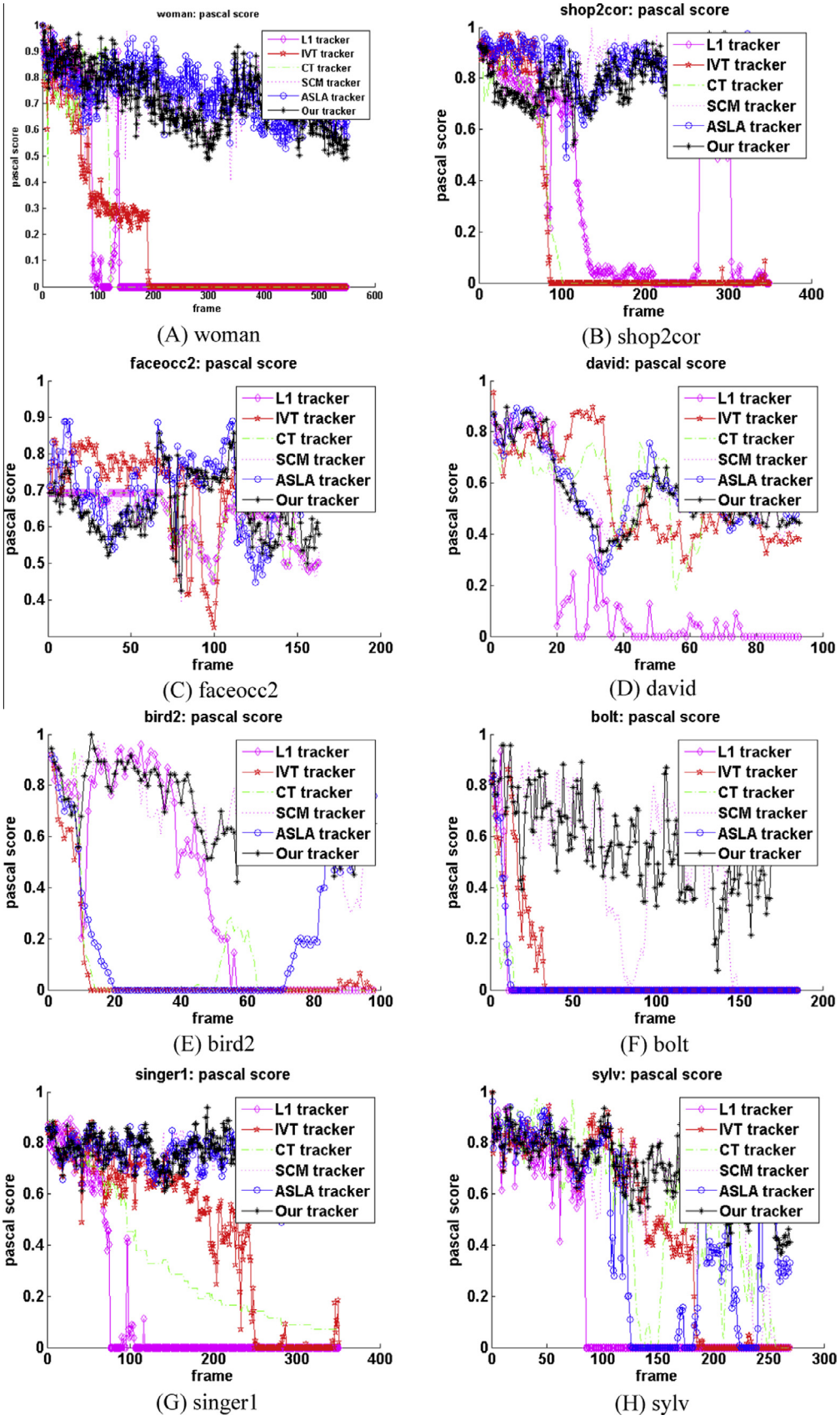


Fig. 6. Pascal score plots for our tracker and the five state-of-the-art trackers.

Table 2
Center location errors (in pixels) of the proposed tracker and the five state-of-the-art trackers. Bold blue font with underline indicates the best performance; bold red font with underline indicates the second best tracker for each sequence.

Video clip	L1 tracker	IVT tracker	CT tracker	SCM tracker	ASLA tracker	Our tracker
woman	141	118	108	4	3	4
shop2cor	23	65	50	3	2	2
faceocc2	13	12	13	6	4	6
david	75	20	15	5	4	5
bird2	151	122	95	11	83	9
bolt	322	49	197	28	97	14
singer1	77	16	21	4	4	3
sylv	91	36	19	15	31	7
Average center location error	111.6	54.7	64.7	9.5	28.5	5.8

Table 3
Pascal scores of the proposed tracker and the five state-of-the-art trackers demonstrate the success rate of the successfully tracked frames for each sequence. Bold blue font with underline indicates the best performance; bold red font with underline indicates the second best tracker for each sequence.

Video clip	L1 tracker	IVT tracker	CT tracker	SCM tracker	ASLA tracker	Our tracker
woman	16.5%	12.2%	20.9%	92.9%	94.9%	84.4%
shop2cor	30.9%	22%	22%	99.4%	99.4%	99.4%
faceocc2	92.0%	91.4%	92.0%	95.1%	96.9%	95%
david	20.4%	37.6%	61.3%	32.3%	34.4%	56.9%
bird2	37.8%	6.1%	9.2%	74.5%	14.3%	80.6%
bolt	6%	5.4%	25.9%	73.1%	12.9%	82%
singer1	19.1%	47.6%	24.2.7%	95.2%	90.0%	99.5%
sylv	35%	51.5%	89.4%	95.5%	96.2%	97.0%
Average Pascal score	32..2%	34.2%	43.1%	82.3%	67.4%	86.9%

42, 72, 128, 154 of bolt sequence, all of the compared trackers except for our tracker cannot handle the cases of very fast motion and severe appearance variations well and thus exhibit severe drift. In general, our tracker achieves the best performance in terms of both accuracy and robustness.

As illustrated in Fig. 4(G), our tracker, SCM tracker and ASLA tracker perform well during the whole sequence, while the L1 tracker cannot handle the sudden illumination changes and drifts from the target in our experiment. Moreover, the trackers of IVT and CT drift away from the target area in frames 197, 314, and 344 due to significant illumination and scale variations.

The last sequence "sylv" is mainly used for the cases of illumination changes and large pose variations. Results on the sequence "sylv" are illustrated in Fig. 4(H). In this sequence, we found that the L1 tracker cannot adapt these changes automatically and thus result in serious drift (see all the frames shown in Fig. 4(H)). Moreover, IVT, SCM and ASLA fail to track the target after frame 924 because of the target appearance change. Fortunately, our tracker can generally handle the appearance change well, yielding a more stable and accurate result than other trackers.

5.2. Quantitative analysis

Two criteria are used to evaluate the performance of the proposed tracker quantitatively. The first one is center location error that measures the Euclidean distance between the central position of the tracking result and that of the manually labeled ground truth. In our experiments, the ground truth centers of the objects in the video clips, woman, shop2cor, faceocc2, david, bird2, bolt, singer1 and sylv, are provided by [13,15,37]. The second one, similar to [34], is success rate that indicates the number of the successful tracked frames and is defined as:

$$Pascalscore = \frac{B_R \cap B_T}{B_R \cup B_T}, \tag{15}$$

where B_R and B_T are the tracked bounding box and the ground truth bounding box, respectively. If the *Pascalscore* is larger than 0.6, it is considered to be successful in tracking for each frame in this paper.

Fig. 5 and Fig. 6 illustrate the comparison of our tracker with the trackers of L1, IVT, ASLA, CT and SCM in terms of center location error and Pascal score. The center location errors and Pascal scores are reported in Table 2 and Table 3, respectively. Experimental results show that our proposed tracking algorithm outperforms the others on the "shop2cor", "bird2", "bolt", "singer1"and "sylv" sequences. As shown in Table 2, our method has the smallest average center location errors implying that it is more robust than the other five trackers. From Table 3, we also observe that our tracker achieves the highest success rate compared with the other five trackers except for the cases of "woman", "david", and "faceocc2" sequences. Moreover, our tracker achieves an average score of

86.9%, which indicates that it is of the highest success rate than the other trackers on the eight experiments.

6. Conclusion

In this paper we propose a novel, robust, and adaptive approach with the design of an collaborative appearance model based on local sparse representation. Different from traditionally local sparse representation model, this work adopts the classification score in SRDM module and the similarity in SRGM module to define a collaborative likelihood measure. Finally, the collaborative likelihood is embedded into a Bayesian inference framework for the estimation of object state in consecutive frames. Our method combines the advantages of generative and discriminative appearance models to account for scene changes. Compared with state-of-the-art tracking methods, the proposed method achieves favorable performance in both the qualitative and quantitative respects.

Acknowledgments

This work was supported by the NSFC-Guangdong Joint Foundation Key Project under Grant (No. U1135003), the National Nature Science Foundation of China (Nos. 61070227 and 61300058).

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2006) 1–45.
[2] Q. Wang, F. Chen, W. Xu, M.-H. Yang, Online discriminative object tracking with local sparse representation. WACV '12 Proc. 2012 IEEE Workshop on the Applications of Computer Vision, January 2012, pp. 425–432.
[3] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 25 (5) (2003) 564–575.
[4] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, Proc. IEEE 98 (6) (2010) 1031–1044.
[5] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparsity-based collaborative model, Comput. Vision. Pattern. Recognit. (2012) 1838–1845.
[6] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 33 (11) (2011) 2259–2272.
[7] B. Liu, J. Huang, et al., Robust tracking using local sparse appearance model and K-selection, Comput. Vision Pattern. Recognit. (2011) 1313–1320.
[8] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, Comput. Vision Pattern. Recognit. (2012) 2042–2049.
[9] X. Jia, H. Lu, M.H. Yang, Visual tracking via adaptive structural local sparse appearance model, Comput. Vision Pattern. Recognit. (2012) 1822–1829.
[10] C. Bao, Y. Wu, H. Ling, H. Ji, Real time robust L1 tracker using accelerated proximal gradient approach, Comput. Vision Pattern. Recognit. (2012) 1830–1837.
[11] S. Avidan, Ensemble tracking, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 29 (2) (2007) 261–271.
[12] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Hengel. A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology (TIST), 4 (4) (2013).
[13] D. Ross, J. Lim, R.-S. Lin, M.-H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vision 77 (1) (2008) 125–141.
[14] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, Comput. Vision Pattern Recognit. (2006) 798–805.
[15] B. Babenko, Y. Ming-Hsuan, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 33 (8) (2011) 1619–1632.
[16] J. Kwon, K.M. Lee, Tracking by sampling trackers, Int. Conf. Comput. Vision (2011) 1195–1202.
[17] S.M.S. Nejhum, J. Ho, M.-H. Yang, Visual tracking with histograms and articulating blocks, Comput. Vision Pattern Recognit. (2008) 1–8.
[18] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) (2008) 210–227.
[19] Z. Han, J. Jiao, B. Zhang, Q. Ye, J. Liu, Visual object tracking via sample-based adaptive sparse representation (AdaSR), Pattern Recognit. (2011) 2170–2183.
[20] Q. Wang, F. Chen, W. Xu, M.H. Yang, Object tracking via partial least squares analysis, IEEE Trans. Image Process. 21 (10) (2012) 4454–4465.
[21] F. Chen, Q. Wang, S. Wang, W. Zhang, W. Xu, Object tracking via appearance modeling and sparse representation, Image Vision Comput. (2011) 787–796.
[22] S.F. He, Q.X. Yang, R. Lau, J. Wang, M.-H. Yang, Visual tracking via locality sensitive histograms, Comput. Vision Pattern Recognit. (2013).
[23] T. Bai, Y.F. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognit. 45 (6) (2012) 2390–2404.
[24] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, Comput. Vision Pattern Recognit. (2013).
[25] Y. Bai, M. TANG, Robust tracking via weakly supervised ranking svm, IEEE Conf. Comput. Vision Pattern Recognit. (2012) 1854–1861.
[26] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognit. 46 (1) (2013) 397–411.
[27] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer, 2001.
[28] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.
[29] K.H. Zhang, L. Zhang, M.H. Yang, Real-time compressive tracking, Eur. Conf. Comput. Vision (2012) 864–877.
[30] <http://www.dabi.temple.edu/~hbling/code_data.htm>.
[31] <http://www.cs.toronto.edu/~dross/ivt/>.
[32] <http://faculty.ucmerced.edu/mhyang/pubs.html>.
[33] R. Cabido, A.S. Montemayor, J.J. Pantrigo, M. Martínez-Zarzuela, B.R. Payne, High-performance template tracking, J. Visual Commun. Image Represent. 23 (2) (2012) 271–286.
[34] M. Everingham, L.V. Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.
[35] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, 1990.
[36] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality constrained linear coding for image classification, Comput. Vision Pattern Recognit. (2010) 3360–3367.
[37] S. Wang, H. Lu, F. Yang, M.-H. Yang, Superpixel tracking, Int. Conf. Comput. Vision (2011) 1323–1330.
[38] G.R. Li, Q.M. Huanga, J.B. Pang, S.Q. Jiang, L. Qin, Online selection of the best k-feature subset for object tracking, J. Visual Commun. Image Represent. 23 (2) (2012) 254–263.