Multimed Tools Appl (2016) 75:8781–8798
DOI 10.1007/s11042-015-2790-3

Robust visual tracking via bag of superpixels

Heng Fan[1] · Jinhai Xiang[2] · Liang Zhao[2]

Received: 24 November 2014 / Accepted: 29 June 2015 /
Published online: 19 July 2015

Abstract The Bag of Words (BoW) model is one of the most popular and effective image representation methods and has been drawn increasing interest in computer vision filed. However, little attention is paid on it in visual tracking. In this paper, a visual tracking method based on Bag of Superpixels (BoS) is proposed. In BoS, the training samples are oversegmented to generate enough superpixel patches. Then *K*-means algorithm is performed on the collected patches to form visual words of the target and a superpixel codebook is constructed. Finally the tracking is accomplished via searching for the highest likelihood between candidates and codebooks within Bayesian inference framework. In this process, an effective updating scheme is adopted to help our tracker resist occlusions and deformations. Experimental results demonstrate that the proposed method outperforms several state-of-the-art trackers.

Keywords Visual tracking · Bag of superpixels (BoS) · Appearance model · *K*-means algorithm

✉ Jinhai Xiang
jimmy_xiang@mail.hzau.edu.cn

Heng Fan
hfan@webmail.hzau.edu.cn

Liang Zhao
zhaoliang323@mail.hzau.edu.cn

[1] College of Engineering, Huazhong Agricultural University, Wuhan 430070, People's Republic of China

[2] College of Informatics, Huazhong Agricultural University, Wuhan 430070, People's Republic of China

1 Introduction

Object tracking is an crucial component of computer vision, such as surveillance, human computer interactions, and human behavior analysis. For adaptive visual tracking, many approaches have been proposed. Despite reasonable good results of these methods, some common challenges remain for tracking objects in complex scenes, e.g., objects suffer from significant pose variations or other severe appearance changes and object pose variations accompanied by long-term object partial occlusions or object intersections [23]. In order to address these problems, a wide range of appearance models have been presented by researchers [17]. Roughly speaking, these appearance models can be categorized into two types: based on holistic representation [3, 18, 20, 28, 29] and based on local representation [2, 11, 15, 24, 27].

The Bag of Words (BoW) image representation [7, 21], which is analogous to the bag-of-words model of text documents [14] in terms of form and semantics, is one of the most effective and efficient image classification methods. The essential idea of this representation is to characterize an image by the histogram of its visual words, that is, vector-quantized local features. Popular candidates for these local features are local descriptor comprising SIFT [8], SURF [4], LBP [19].

Motivated by the BoW model, a novel tracking method based on Bag of Superpixels (BoS) is proposed in this paper. In traditional BoS, the local patches are based on low-level pixels. In visual tracking, the object appearance changes a lot due to many factors, e.g., occlusion, deformation, illumination variation, pose change. When object appearance changes, the pixels are prone to be affected. Thus these low-level based features are not stable, which easily causes drift problems even tracking failure. To account for large appearance in tracking, it is of great interest to develop adaptive appearance model based on mid-level visual cues. In the superpixels, the similar pixels are grouped into one superpixel. When the object appearance varies, single pixel is easily affected, but the superpixel is stable as a whole. Therefore our approach utilizes the superpixel patches to substitute the low-level local patches to construct the codebook. In our BoS model, the training samples obtained are oversegmented into superpixels. In order to generate superpixel patches for training, a rectangle grid is superimposed on the object superpixel map to segment the target into uniform blocks. In the rectangle, each block is represented by its center point, which is annotated to the feature vector of the covered superpixel. Hence the object is represented by a collection of center points, which are associated to the superpixel patches. After obtaining the patches, we perform K-means algorithm on these patches to form visual words of the target and construct a superpixel codebook for the object. The tracking is formulated as searching for the highest likelihood between candidate objects and codebooks within the Bayesian framework. In addition, an effective update scheme is adopted to adapt our tracker to the occlusion and deformation.

Through the above analysis, a BoS tracking method is proposed in this papers. The contributions of this work can be summed up in three aspects:

- First, a mid-level based appearance model BoS is presented for visual tracking. The BoW is an effective image categorization model, which extracts local features of the object in low-level. In visual tracking, however, the object often suffers from appearance variations. To handle this issue, we exploit mid-level visual cues for adaptive appearance model and propose a robust BoS for object tracking.
- Second, we modify the traditional BoW to accommodate it to object tracking. In the traditional BoW, for example, the local patches are extracted randomly, which is not

suitable for visual tracking. In visual tracking, the object appearance is prone to be affected by many factors, such as occlusion, deformation, etc. In these cases, the local patches extracted for appearance modeling should cover the whole object and be able to discriminative the object. If we extract patches randomly, these patches may be all affected by appearance changes and not discriminative to distinguish the object from the background. For example, when occlusion occurs, the randomly extracted patches may be occluded and invalidated to recognize the target, which results in drift to the background. In our BoS, a rectangle grid is superimposed on the object superpixel map to segment the target into uniform patches, which guarantees that enough and rich local information of the object can be obtained. Besides, the similarity of a patch with the codeword in the codebook is used to provide more information. In contrast, the traditional BoW and many improvements only care the similarity between histograms formed by calculating the frequency of occurrence of codewords in images, discarding the similarities of patches. The experiments demonstrate that the proposed BoS is effective for visual tracking.

- Third, a novel online update method is proposed in this paper to help the tracker adapt to the appearance variations. The proposed tracker uses the old codebook and new feature collection to update the object appearance model, and an oblivious factor is used to control the update. Different from other trackers, the oblivious factor is dynamic and relies on the change of object appearance. If the object suffers from occlusion, the new features collected are bad, and thus the value of oblivious factor is large, which means that the old codebook is paid more attention; otherwise the value of oblivious factor is small. Therefore, our tracker can keep the object appearance effective.

The rest of the paper is organized as follows. Section 2 briefly reviews the related works. Section 3 simply describes the standard BoW algorithm. We give the details of our BoS model in Section 4. Section 5 introduces the proposed tracking method. Experimental results are shown in Section 6, and Section 7 concludes this paper.

2 Related works

Visual tracking methods can be roughly categorized into either holistic based representation or local based representation. Appearance models based on global representation reflect the global statistical characteristics of object appearance. In general, the global visual representations are simple and computationally efficient for fast object tracking. Due to the imposed holistic geometric constraints, the global models are susceptible to global appearance variations (e.g., caused by illumination variation, pose change or partial occlusion). To handle complex appearance changes, a variety of methods have been proposed. In [20], the incremental visual tracking method suggests an online approach for efficiently learning and updating a low dimensional PCA subspace representation for the object. However, this PCA subspace based representation scheme is sensitive to partial occlusion. Babenko et al. [3] utilize the multiple instance learning (MIL) method for visual tracking, which can alleviate drift to some extent. Whereas the MIL tracker may detect the positive sample that is less important because it does not consider the sample importance in its learning process. Further, Zhang et al. [28] propose the online weighted multiple instance learning (WMIL) by assigning weight to different samples in the process of training classifier. Nevertheless this approach easily results in drifting problem or tracking failure because it is much dependent on global template information of the object, which is likely occluded. Mei et al. [18]

achieve tracking via modeling object appearance by sparse representation, and Zhang et al. [29] adopt compressive sensing to locate object. However these approaches still can not address the problems caused by appearance changes.

In contrast, the local based representations are able to capture the local structure of the object. Consequently, the local appearance models are robust to global appearance variations caused by shape deformation, illumination variation, partial occlusion and rotation. Nevertheless most local visual representations, which are typically required by discriminative feature selection, need a huge number of local features (e.g., color, texture and shape), resulting in a very high computational cost. Adam et al. [2] propose a fragments-based tracking approach, and further Wang et al. [24] embed the fragments-based method into mean shift tracking framework. These tracking methods estimate the target according to the voting map of each part via comparing histogram between each part and corresponding template. Nevertheless, static template with equal importance being assigned to each fragment greatly lowers the performance of tracker. Kalal et al. [15] propose a algorithm called P-N learning, which collects local features from positive and negative samples, to construct a classifier by learning form these features. This tracker easily leads to drift problem as the discriminative ability of the classifier is gradually undermined without considering the importance of samples. Fan et al. [10] suggest a part-based visual tracking algorithm via weighted P-N learning. This method improves the robustness of the tracker. Yang et al. [27] introduce a superpixel tracking method. This method only calculates the probabilities of superpixels belonging to foreground, which easily leads to drift in color-similar background and whose tracking results will shrink to the unoccluded part of object in the presence of occlusion. Fan et al. [11] suggests a structural superpixel graph tracking method to address this problem.

Recently, the local-based representation Bag of Words (BoW) has been widely used in image classification, shape representation, action recognition and object recognition due to its simplicity and robustness. Many researches based on BoW have been reported. Fei-Fei et al. [16] use BoW to learn natural scene categories by modeling the images as a collection of codewords and construct a Bayesian hierarchical model to describe the codewords distribution. Sivic et al. [22] and Fergus et al. [12] utilize the similar methods for image classification. Wang et al. [25] develop a new shape representation called Bag of Contour Fragments (BCF) inspired by classical Bag of Words model. Iosifidis et al. [13] apply BoW model into human action recognition. Bolovinou et al. [5] combine BoW with spatial context and propose a Bag of Spatio-Visual Words (BoSVW) representation for scene classification. Extensive experiments demonstrate that BoW has good performance to deal with intra-class pose variant and occlusion in image classification and recognition. It can successfully identify a right image, only relying on a specific set of representative features.

The most related works to ours are [27] and [26]. In [27], a superpixel based tracking is proposed, in which the object area (containing both object and background) is oversegmented into superpixels, and the object and background is directly represented with the superpixel cluttering centers. A target-background classifier is constructed based on these object and background cluttering centers. The object tracking is formulated as searching for the candidate containing most target superpixels on a confidence map. However, the proposed method is different from [27] although superpixel is utilized as well. In our work, only target information is utilized. The object is viewed as a bag and oversegmented into superpixels. These superpixels are grouped into clusters to form superpixel codewords and construct the codebook, and visual tracking is accomplished by finding the candidate with the highest similarity with codebook. In [26], Yang et al. present a tracking approach using bag of features. They directly extract local patches of the object to generate codewords and apply BoW model to constructing the codebook. Nevertheless, our method is different from [26] in two aspects. First, we use local superpixel patches based on mid-level to form codewords and develop a superpixel codebook instead of utilizing local patches based on low-level. It is of great advantages to develop an adaptive appearance model using mid-level visual cues. Second, a grid rectangle is utilized to cover the object superpixel map, with which we can extract enough and rich local superpixel patches. In [26], however, they just extract local patches at random, which cannot guarantee that the codebook constructed contains enough object information. Especially, the tracker easily drifts in the presence of occlusion because the local patches extracted from the object are occluded. Besides, the online update method in this paper is different from [27] and [26]. We use the latest feature collection and old codebook for update, and a dynamic oblivious factor is adopted to control the update. When the object suffers from occlusions, the old codebook is paid more attention, otherwise the new feature collection is assigned with more importance. In conjunction with our online update, the proposed tracker is more robust to occlusion.

3 Standard bag of words for image classification

Standard BoW algorithm for image classification comprises four steps: feature extraction, bag construction, image representation and classification. In [16], a hierarchical model for learning natural scene categories is proposed, as shown in Fig. 1. For training images

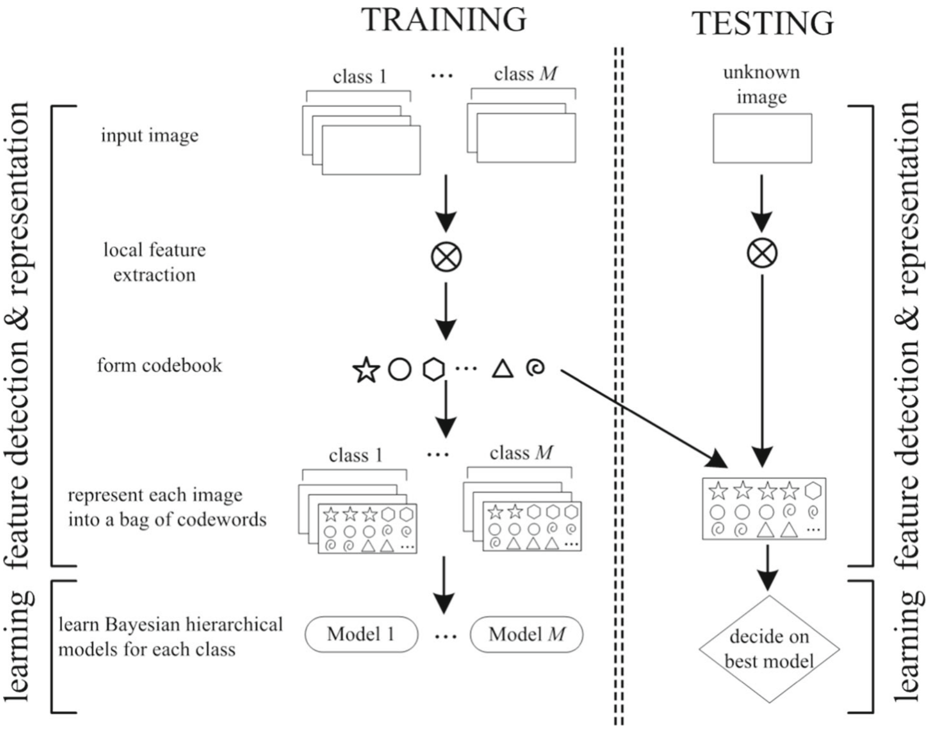

TRAINING
TESTING
class 1
class *M*
unknown image
input image
local feature extraction
form codebook
class 1
class *M*
represent each image into a bag of codewords
learn Bayesian hierarchical models for each class
Model 1
Model *M*
decide on best model
feature detection & representation
learning
feature detection & representation
learning

Fig. 1 Bag of Words model for image classification in [16]

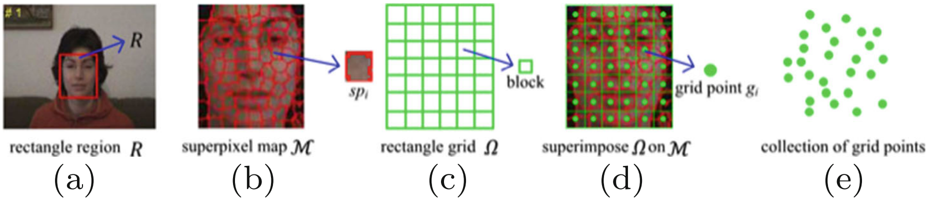

#1
R
rectangle region R
(a)
superpixel map $\mathcal{M}$
(b)
sp_i
block
rectangle grid Ω
(c)
grid point g_i
superimpose Ω on $\mathcal{M}$
(d)
collection of grid points
(e)

Fig. 2 The process of obtaining superpixel patches. The *solid red rectangular* region R in image (**a**) is the target. Image (**b**) is the superpixel map $\mathcal{M}$, which is composed of superpixel patches. Image (**c**) shows the rectangle grid, which consists of blocks. In image (**d**), the green rectangle grid Ω is imposed on superpixel map $\mathcal{M}$. The *solid green dot* g_i in each block denotes the grid point, and each grid point g_i is covered by a superpixel patch sp_i. Image (**e**) shows the collection $G = \{g_i\}_{i=1}^{N}$ of grid points, which represent the superpixel patches

belonging to M classes, a detector is used to obtain scale and rotation invariant interest points, where the image patches around the points with varied scale and rotation are gathered. Then the descriptors are used to describe these local patches. The effective candidate descriptors for these features are called local descriptors, such as SIFT, SURF, LBP and HSI.

Given a set of features, codebook is formed through K-means algorithm, where K is the number of clutters, as well as the size of codebook. The centres of clusters are the codewords. Thus descriptors in each training image can be coded by hard assignment to the nearest codeword, generating a histogram $n(w_i, c_j)$ counting the frequency of the occurrence of each codeword, where w_i is the i^{th} word in the codebook and c_j denotes j^{th} class. The histogram is treated as a bag. For a test sample, it can be represented with a histogram, and the corresponding image class can be determined by searching for the most similar training histogram.

4 Bag of superpixels (BoS)

4.1 Superpixel feature extraction

In this paper, we utilize superpixel patches to represent the target. For the rectangle region R in one frame (See Fig. 2a), firstly we segment it into superpixels via SLIC [1] and obtain the object superpixel map $\mathcal{M}$ (See Fig. 2b). A rectangle grid Ω (See Fig. 2c) is then employed to superimpose on $\mathcal{M}$, which divides the target into several uniform blocks (See the green blocks in Fig. 2d). In our work, we divide the rectangle grid from the superpixel map instead of the image target because the local patches obtained directly from the target are pixel based, which are low-level cues and not stable for visual tracking. Therefore we divide the rectangle grid from the superpixel map and obtain the superpixel patches.

Assume that there are N blocks. For each block, it is represented by its center point (See the green solid dots in Fig. 2d). Let g_i denote the center, and the collection of these center points is denoted as $G = \{g_i\}_{i=1}^{N}$ (See Fig. 2e). Suppose that the relative position of g_i in the superpixel map is p_i, and sp_i represents the superpixel patch which covers the grid point g_i at location p_i on the superpixel map. For each superpixel patch sp_i, its feature vector (e.g., HSI histogram) is denoted by f_i. In this paper, the feature vector of the grid point g_i is represented with the feature vector of sp_i, and feature vectors of the grid points is denoted by the set $F = \{f_i\}_{i=1}^{N}$.

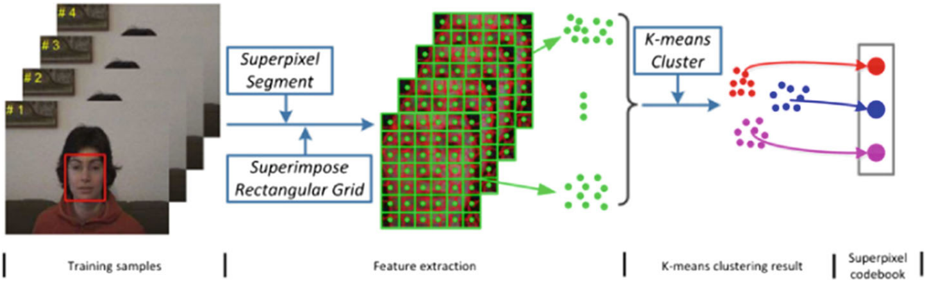


Fig. 3 Illustration of the superpixel codebook construction

4.2 Superpixel codebook construction

Before tracking, adequate superpixel patches are needed for constructing the superpixel codebook. To provide enough training samples, incremental visual tracking (IVT) [20] is adopted to track the first Q frames (typically four frames) because the object hardly suffers from any appearance variations in these frames, and IVT tracker can accurately locate the object in this situation. Besides, the IVT tracker is highly efficient, which means that it has a very low time and computational cost. Therefore we choose IVT tracker to track the object in the first Q frames and collect enough training samples. In each training frame, we extract N superpixel features to represent the object. After obtaining the features, the codewords are generated by K-means clustering algorithm.

K-means algorithm is taken into our consideration because it has several advantages [6]: (1) the time and storage complexity are both linear with respect to the data points; (2) it is guaranteed to converge at a quadratic rate; (3) it is invariant to data ordering. The time and storage complexity is the most fundamental factor that needs to be considered in classification because massive data are involved in the large scale application circumstances. Through performing K-means clustering algorithm, these features are grouped into K clusters. The codewords of the superpixel codebook is defined as the clutter centers. Figure 3 illustrates the construction process.

After superpixel codebook construction, training samples are represented by superpixel codewords as bags. A bag is equal to the occurrence frequency of codewords in an image and can be represented as a histogram. Q training images are converted to a set of bags $\{B_q\}_{q=1}^{Q}$ via raw counts.

4.3 Similarity of BoS

In image classification, a crucial aspect is to measure the similarity between the candidate image and the reference classes. In BoS, the similarity between the candidate and the codebook is decomposed into bag similarity and patch similarity. Let $\{C_k\}_{k=1}^{K}$ be the superpixel codebook, S denotes the candidate image. For the codebook, the feature of each codeword C_k is represented with f_k^c. For S, we extract N superpixel features according to Section 3.1, which is represented by a set $F^s = \{f_i^s\}_{i=1}^{N}$. We use B_s to denote the bag of candidate image.

For the candidate, the bag similarity with the q^{th} bag is defined as Γ_q^B via

$$\Gamma_q^B = exp(-dist(B_q, B_s)) \tag{1}$$

where B_q and B_s are the bags of the q^{th} training image and candidate respectively, and $dist$ is a function for measuring the distance between two bags via χ^2 test. It is worth noting that the bags are statistic features, thus χ^2 test is more suitable for measuring the similarity between bags.

The distance $d_{i,k}$ between i^{th} $(i = 1, 2, \cdots, N)$ superpixel feature f_i^s of candidate and k^{th} $(k = 1, 2, \cdots, K)$ codeword C_k in codebook can be obtained by

$$d_{i,k} = exp\left(-\| f_i^s - f_k^c \|_2^2\right) \tag{2}$$

where f_i^s is the i^{th} feature vector of candidate, and f_k^c represents the feature vector of the k^{th} codeword in codebook. Let D_i be the minimum distance between the i^{th} feature and codebook, and it is defined as

$$D_i = \min_{\{C_k\}_{k=1}^K} d_{i,k} \tag{3}$$

We use $I_i \in [1, K]$ to record index of the codeword, which has the minimum distance with the i^{th} feature. The I_i can be obtained via

$$I_i = \arg\min_k \ d_{i,k} \tag{4}$$

After obtaining the I_i, we can compute the bag B_s of the candidate. For i from one to N, we can employ (2)–(4) to compute B_s via

$$B_s(I_i) = B_s(I_i) + 1 \quad i = 1, 2, \cdots, N \tag{5}$$

where B_s is the histogram representing the bag which is initialized to a zero vector with length K, and $B_s(I_i)$ is the I_i bin of the histogram. Thereby the bag similarity Γ^B of the candidate is determined through

$$\Gamma^B = \max_{\{B_q\}_{q=1}^Q} \Gamma_q^B = \max_{\{B_q\}_{q=1}^Q} exp(-dist(B_q, B_s)) \tag{6}$$

and its patch similarity Γ_t^P can be gotten by

$$\Gamma^P = \sum_{i=1}^{N} exp(-D_i) \tag{7}$$

In our work, the similarity between the candidate and the codebook is composed of bag similarity and patch similarity. The bag similarity is computed by the bag histograms, which measures the holistic features of the candidate and the target. However, the patch similarity is calculated by the feature vectors of the local patches, which measures the local features of the candidate and the target. These two similarity compare the similarity between the candidate and the codebook from two different views and are equally important for measuring the similarity. Therefore, the combined similarity Γ is obtained via

$$\Gamma = \Gamma^B + \Gamma^P \tag{8}$$

The similarity Γ indicates how likely the candidate belongs to the reference class, which is represented by the codebook.

5 Visual tracking via BoS

In this section, the proposed BoS tracker is introduced. Section 5.1 describes the tracking formulation in this work, and an online dynamic updating mechanism is presented in Section 5.2.

5.1 Tracking formulation

Our tracker is implemented via the Bayesian framework. Given the observation set of target $Y^t = \{y_1, y_2, \cdots, y_t\}$ up to the frame t, we can obtain estimation $\widehat{X}_t$ by computing the maximum a posterior via

$$\widehat{X}_t = \arg\max_{X_t^i} p(X_t^i|Y^t) \tag{9}$$

where $\widehat{X}_t$ denotes the i-th sample at the state of X_t. For X_t, its state $\widehat{X}_t$ is defined by $\widehat{X}_t = (c_x, c_y, c_s)$, where (c_x, c_y) is the centroid position and c_s is its scale. The posterior probability $p(X_t^i|Y^t)$ can be obtained by the Bayesian theorem recursively via

$$p(X_t|Y^t) \propto p(y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y^{t-1}) dX_{t-1} \tag{10}$$

where $p(X_t|X_{t-1})$ and $p(X_{t-1}|Y^{t-1})$ represent the dynamic model and observation model respectively.

The dynamic model indicates the temporal correlation of the target state between consecutive frames. We apply affine transformation to modeling the target motion between two consecutive frames within the particle filter framework. The state transition can be formulated as

$$p(X_t|X_{t-1}) = N(X_t; X_{t-1}, \Psi) \tag{11}$$

where Ψ is a diagonal covariance matrix whose elements are the variance of affine parameters. The observation model $p(y_t|X_t)$ represents the probability of the observation y_t as state X_t. In this paper, the observation is designed by

$$p(y_t|X_t) \propto \Gamma_t \tag{12}$$

where the right side of the equation denotes the similarity between the t^{th} candidate and the codebook, which can be obtained based on Section 4.3. With the help of updating strategy, the observation model can robustly adapt to the appearance changes of object.

5.2 Online update

Due to the appearance variations of target, updating is essential. In this paper, we propose an effective superpixel codebook updating mechanism.

After tracking the target in each frame, we sort the associated superpixel features decreasingly according to the values $\{exp(-D_i)\}_{i=1}^N$, which indicates the similarity between a superpixel feature and the codebook, and collect p superpixel features with maximum value $exp(-D_i)$. Repeating the sorting and collecting in the next γ frames, a new feature collection $\{f_i'\}_{i=1}^{p\cdot\gamma}$ can be obtained. K-means clustering algorithm is performed again on the new collection $\{f_i'\}_{i=1}^{p\cdot\gamma}$ and old codebook $\{C_k\}_{k=1}^K$ through

$$\{C_k'\}_{k=1}^K = kmeans((1-\lambda)\{f_i'\}_{i=1}^{p\cdot\gamma}, \lambda\{C_k\}_{k=1}^K) \ (0 < \lambda < 1) \tag{13}$$

where $\{C_k'\}_{k=1}^K$ is the new superpixel codebook and λ is an oblivious factor to control the update.

In our paper, the value of λ is dynamic and relies on the superpixel features obtained, which is different from [26] in which the oblivious factor λ is fixed in all the situation and they ignore the weight of the new feature collection. If the object suffers from large appearance variation caused by occlusion, the value of λ should be large, which means that the old codebook is more important for update; otherwise λ should be small, which signifies that the new feature collection should be paid more attention. The λ is determined by

$$\lambda = \frac{1}{\gamma}\sum_{i=1}^{\gamma}\frac{\varphi_i}{p} \tag{14}$$

where γ denotes the number of frames stored for update, p represents the number of superpixel patches collected in each stored frame and φ_i is the number of bad superpixel patches collected in the i-th ($i = 1, 2, \cdots, \gamma$) frame. The more the bad patches are, the large the value of λ is, which means that the object appearance changes a lot. The φ_i can be computed with

$$\varphi_i = \sum_{j=1}^{p} b_j \tag{15}$$

where b_j is used to describe the j-th ($j = 1, 2, \cdots, p$) patch. If b_j is 1, the j-th patch is bad; otherwise the j-th patch is good. The b_j depends on the minimum distance between the j-th patch and the codebook and is obtained by

$$b_j = \begin{cases} 1, & D_j > D_{th} \\ 0, & Otherwise \end{cases} \tag{16}$$

where D_j represents the minimum distance between the j-th patch and the codebook, and D_{th} is a predefined threshold parameter.

Through the proposed update method, when occlusion occurs during tracking, the bad superpixel patches collected will increase, and thus the value of λ is large, which means that the old codebook is paid more attention. If the target does not undergoes occlusion, the number of bad superpixel patches is small, and thus λ is small, which signifies that the new feature collection is assigned with more importance. Therefore, our tracker can keep the object appearance effective.

So far, we have introduced the overall procedure of the proposed tracking algorithm as shown in Algorithm 1.

Algorithm 1 BoS Tracking

```
Initialization:
Select initial object region $R$ and segment it into superpixels;
Track the first $Q$ frames with IVT tracker;
Extract superpixel features for the tracked targets in the first
$Q$ frames according to Section 4.1;
Perform $K$-means clustering algorithm on obtained features and
construct the superpixel codebook with a codeword set $\{C_k\}_{k=1}^{K}$;
Convert training samples to a set of bags $\{B_q\}_{q=1}^{Q}$;
Tracking:
for $t = Q + 1$ to the end of the sequence do
    Generate $L$ candidate targets via the Bayesian framework;
    for $l = 1$ to $L$ do
        Extract superpixel features for the $l^{th}$ candidate
        according to Section 4.1;
        Compute the similarity $\Gamma_l$ for each candidate;
    end for
    $\mathcal{L} = \mathrm{argmax}_l\, \Gamma_l$;
    The $\mathcal{L}^{th}$ candidate is the tracked object;
    Update superpixel codebook according to Section 5.2;
end for
End
```

Table 1 Image sequences used in our experiments

Sequence	Frames	Main Challenges	Resolution
Basketball	725	Occlusion, Deformation, Background clutter	High
Cup	303	Background clutter, Scale variation	Low
Face	415	Occlusion	High
Jogging	307	Deformation, Occlusion	High
Woman	551	Occlusion, Deformation, Scale variation	High
Bird2	99	Occlusion, Scale variation, Deformation	High
Bicycle	271	Occlusion, Scale variation	High
Bolt	350	Deformation, Scale variation, Background clutter	High

6 Experimental results

In order to evaluate the performance of our tracking algorithm, we test our tracker on eight challenging image sequences. All the initial targets in the sequences are selected manually in the first frame. These sequences comprise most challenging situations in object tracking as shown in Table 1. Further, we compare our tracker with six state-of-the-art tracking algorithms. These algorithms are Frag tracking [2], TLD tracking [15], ℓ_1 tracking [18], IVT tracking [20], MIL tracking [3] and CT tracking [29]. Some representative results are displayed in this section.

The proposed algorithm is implemented in MATLAB on a 3.2 GHz Intel E3-1225 v3 Core PC with 8GB memory. The parameters of the proposed tracker are fixed in all experiments. The number of particles in Bayesian framework is 300. The training frame Q is 4 and the size of the codebook K in this work is set to 20. The parameter γ is fixed to 5 in updating The predefined threshold parameter D_{th} is set to 0.8.

6.1 Quantitative comparison

We evaluate the above mentioned trackers via center location error and overlapping rate [9], and the comparing results are shown in Tables 2 and 3. Figure 4 shows the center location error of the trackers on eight test sequences. Overall, the tracker proposed in this paper outperforms the state-of-the-art algorithms. Besides, we report the computational cost of the proposed method in Table 2 as well and compare it with other algorithms. The computational cost in this paper is measured by frame per second (FPS). From the Table 2 below, we can see that the average FPS of the proposed tracker is 1.2, which is slower than most state-of-the-art tracking methods (except for ℓ_1), because our tracker needs to segment the object into superpixels to obtain mid-level visual cues. However, the comprehensive tracking accuracy of our tracker is higher than other methods.

6.2 Qualitative comparison

Heavy occlusion The targets in the sequence *Face*, *Woman*, *Jogging* and *Bicycle* undergo severe occlusion as shown in Fig. 5. Owing to the occlusion, the object appearance changes a lot. Thus an effective appearance model and update scheme are essential. In IVT and CT tracking, the targets are prone to drift away because they do not have any mechanism to resist occlusion. Although Frag, TLD, MIL and ℓ_1 are robust under occlusion, they still do not

Table 2 Center location errors (CLE) (in pixels) and average frames per second (FPS). The best result is shown in **red** and the second best in **blue** font

Sequence	Frag	TLD	ℓ_1	IVT	MIL	CT	Ours
Basketball	**16.25**	129.37	131.25	68.42	94.83	19.02	**5.22**
Cup	7.07	3.12	2.92	**1.81**	40.60	25.13	**2.73**
Face	**4.86**	6.91	5.44	62.55	36.22	57.07	**4.68**
Jogging	9.3	**7.2**	14.59	130.00	146.18	124.76	**10.32**
Woman	106.21	**72.63**	134.15	138.64	116.35	108.88	**8.23**
Bird2	50.04	74.86	66.72	58.76	**14.83**	22.05	**14.55**
Bicycle	55.77	10.63	49.03	**7.82**	10.5	51.5	**5.7**
Bolt	100.6	**87.85**	361.79	374.95	365.4	348.8	**10.7**
CLE	**43.77**	49.07	95.74	105.37	103.11	94.65	**7.77**
FPS	4	18	0.5	32	32	90	1.2

perform well because their update strategies mistake the background for target and update, which causes tracking drift. In our method, BoS tracker collect enough local features to model the target. When occlusion happens, there are parts of the object still unoccluded, our tracker can capture these unoccluded local features and utilize them to distinguish the object from the background. Besides, our update scheme can help the proposed tracker obtain the latest local features. Therefore, our method can robustly locate the target and achieve the best tracking performance even in the presence heavy occlusions occur.

Deformation Deformation is a disaster for tracker, because the object appearance have changed significantly when deformation happens. As shown in Fig. 5, ℓ_1 , IVT, CT do not have good performances in the sequence *Basketball*, *Bird2*, *Jogging* and *Bolt* because these trackers are based on holistic representation. When deformation happens, the holistic representation based models are varied and cannot recover from the deformation. Differently, TLD, Frag and MIL have relatively better tracking results in these sequences, however, these methods still fail in sequences Woman because deformation and occlusion occur at

Table 3 Overlapping rate (OR) (in pixels). The best result is shown in $_{\mathrm{a}}$nd the second best in **red** font

Sequence	Frag	TLD	ℓ_1	IVT	MIL	CT	Ours
Basketball	0.55	0.09	0.03	0.41	0.21	**0.61**	**0.72**
Cup	0.67	0.72	**0.74**	0.71	0.39	0.53	**0.77**
Face	**0.86**	0.77	0.85	0.36	0.55	0.38	**0.87**
Jogging	**0.65**	0.73	0.57	0.13	0.01	0.13	**0.64**
Woman	0.19	**0.29**	0.06	0.16	0.13	0.17	**0.59**
Bird2	0.37	0.22	0.40	0.39	**0.60**	0.48	**0.71**
Bicycle	0.25	0.39	0.31	0.33	**0.43**	0.29	**0.65**
Bolt	**0.2**	0.14	0.02	0.01	0.01	0.01	**0.62**
OR	**0.47**	0.42	0.37	0.31	0.29	0.33	**0.69**

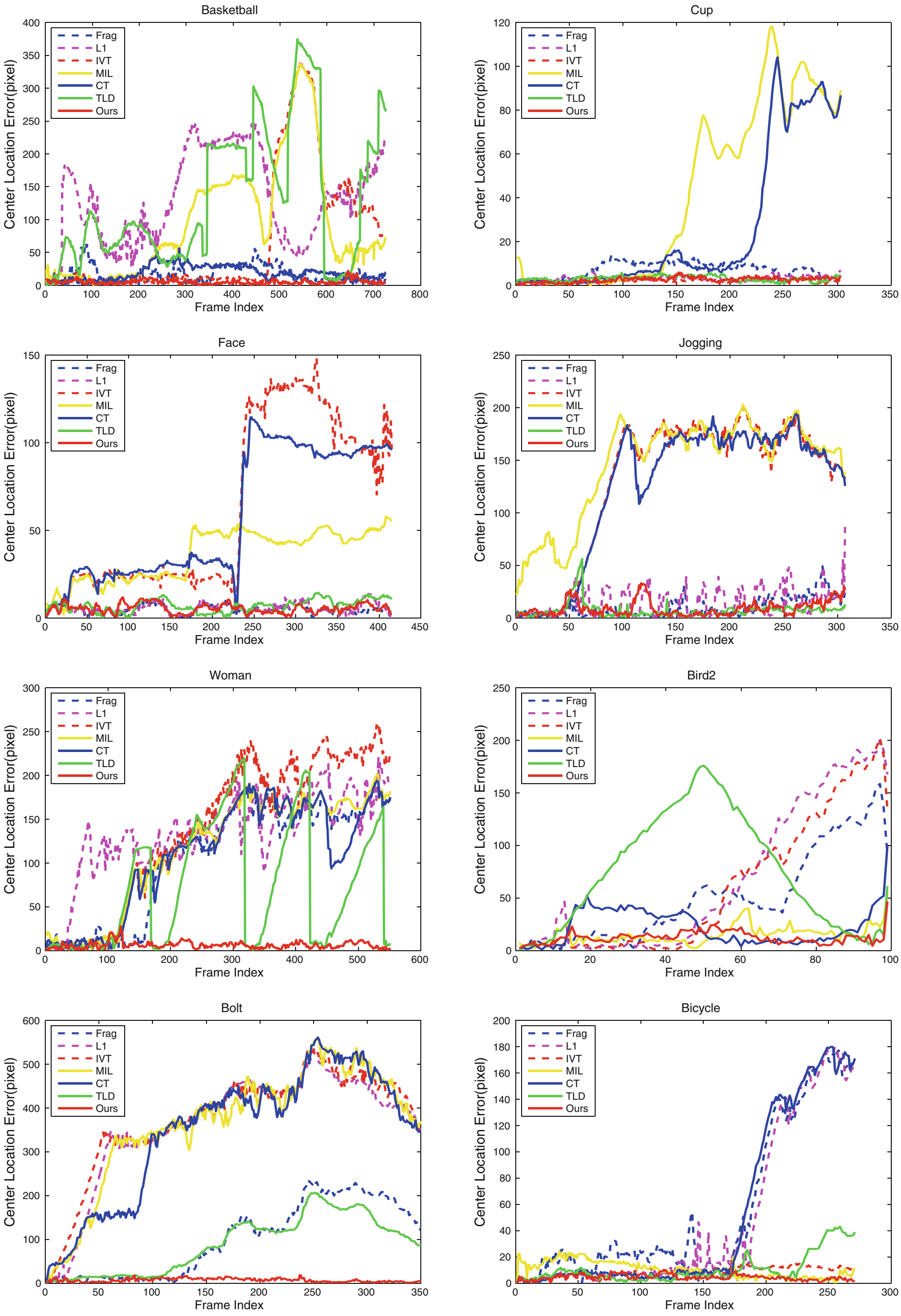
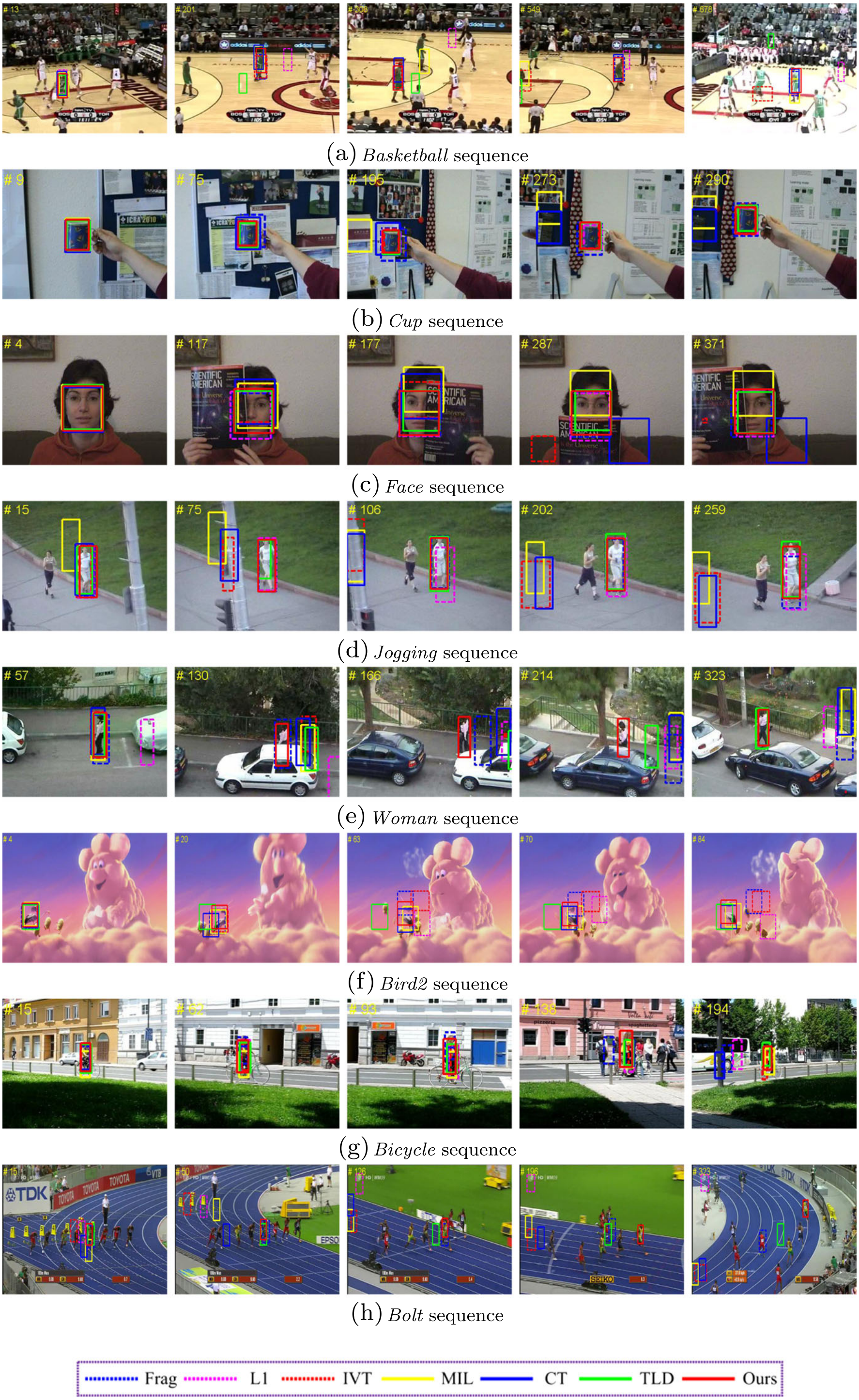


Fig. 4 Quantitative evaluation in terms of center location error (in pixel). The proposed method is compared with six state-of-the-art algorithms on six challenging test sequences



(a) *Basketball* sequence

(b) *Cup* sequence

(c) *Face* sequence

(d) *Jogging* sequence

(e) *Woman* sequence

(f) *Bird2* sequence

(g) *Bicycle* sequence

(h) *Bolt* sequence

Fig. 5 Tracking results on various challenging sequences

the same time. Our tracker performs well in handling structural deformation. Because our appearance model extracts the enough and rich local features of the object in mid-level, and these features are orderless and not sensitive to deformation. When deformation and occlusion occur simultaneously, our tracker can grasp the local stable feature to discriminate the target and background.

Pose variation The image sequences *Bird2* and *Basketball* suffer from pose variation during the tracking. The object appearance changes thoroughly in the presence of pose changes. As shown in Fig. 5, the Frag, ℓ_1 , IVT, MIL, CT and TLD because these trackers cannot effectively update the appearance model. For IVT, MIL, ℓ_1 and CT trackers, these methods are based on holistic representations. When updating appearance, the background is added into the appearance, which undermines the performance of these trackers. For TLD and Frag, these approaches are local based representations. Nevertheless, their update schemes cannot adapt their appearance models to the pose variation. In out method, we collect good tracking results every frames and extract local features from them, which can make our tracker resist to the pose variation effectively.

Background clutter The sequence *Cup*, *Basketball* and *Bolt* in Fig. 5 suffer from background clutter. The traditional BoW randomly extract local features from the object, which are not enough for visual tracking. When background clutter and occlusion occur at the same time (e.g., Bolt and Basketball), the random local features might be occluded. In this case, the BoW model cannot recognize the object. Whereas our BoS model extracts rich and enough local features. When background clutter and occlusion occur at the same time, our tracker are still able to locate the object from the background clutter by using the unoccluded local information. Besides, the updating scheme also helps our tracker robustly locate the target in complex background.

7 Conclusion

In this paper, we propose a BoS tracking framework. The object is represented with a set of center points, which are related to local superpixel patches, and each point is annotated to a feature vector. Through incremental subspace learning tracking method, we collect enough training samples to construct superpixel codebook by K-means algorithm. Meanwhile, we design a updating scheme to obtain the latest superpixel codewords for updating the codebook. Experiments demonstrate that our approach outperforms several state-of-the-art trackers.

Although the proposed tracker performances well on various challenging sequences, the computational cost is high because our tracker needs to segment the object into superpixels at the beginning, which is computationally complex. One potential solution is to optimize the clustering algorithm in superpixel segmentation in which the initial cluster centers are selected at random, and the new cluster centers are computed by iteration. If the cluster centers can be optimized initially by an effective algorithm, the superpixel segmentation will be more efficient and our tracker thus is much faster. Therefore, the main future work of this paper is how to reduce the computational complexity via optimizing clustering algorithm in superpixel segmentation.

Acknowledgment Jinhai Xiang is supported by the Fundamental Research Funds for the Central Universities (Program No.2014BQ083). Liang Zhao is supported by the Fundamental Research Funds for Central Universities (Program No.2014QC008).

References

1. Achanta R, Shaji A, Smith K et al (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
2. Adam A, Rivlin E, Shimshoni I (2006) Robust fragments-based tracking using the integral histogram. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 798–805
3. Babenko B, Yang M-H, Belongie S (2011) Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell 33(8):1619–1632
4. Bay H, Ess A, Tuytelaars T et al (2008) Speeded-up robust features (SURF). Comput Vis Image Und 110(3):346–359
5. Bolovinou A, Pratikakis I, Perantonis S (2013) Bag of spatio-visual words for context inference in scene classification. Pattern Recogn 46(3):1039–1053
6. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40(1):200–210
7. Dance C, Willamowski J, Fan L et al (2004) Visual categorization with bags of keypoints. In: European conference on computer vision workshop on statistical learning in computer vision (ECCVW), pp 59–74
8. David GL (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
9. Everingham M, Gool LV, Williams CKI et al (2010) The PASCAL visual object classes challenge. Int J Comput Vis 88(2):303–338
10. Fan H, Xiang JH, Xu J et al (2014) Part-based visual tracking via online weighted P-N learning. Sci World J 2014:13
11. Fan H, Xiang JH, Liao HH et al (2015) Robust tracking based on local structural cell graph. J Vis Commun Image R 31:54–63
12. Fergus R, Li F-F, Perona P et al (2005) Learning object categories from Google's image search. In: IEEE international conference on computer vision (ICCV), pp 1816–1823
13. Iosifidis A, Tefas A, Pitas I (2014) Discriminant bag of words based representation for human action recognition. Pattern Recogn Lett 49(1):224–230
14. Joachims T (1998) Text categorization with suport vector machines: learning with many relevant features. In: European conference on machine learning (ECML), pp 137–142
15. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell 34(7):1409–1422
16. Li F-F, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 524–531
17. Li X, Hu W, Shen C et al (2013) A survey of appearance models in visual object tracking. ACM Trans Intel Syst Tec 4(4):2411–2418
18. Mei X, Ling HB (2011) Robust visual tracking and vehicle classification via sparse representation. IEEE Trans Pattern Anal Mach Intell 33(11):2259–2272
19. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
20. Ross D, Lim J, Lin RS et al (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1):125–141
21. Sivic J, Zisserman A (2003) Video google: A text retrieval approach to object matching in videos. In: IEEE international conference on computer vision (ICCV), pp 1470–1477
22. Sivic J, Russell BC, Efros AA et al (2005) Discovering objects and their location in images. In: IEEE international conference on computer vision (ICCV), pp 370–377
23. Tsagkatakis G, Savakis A (2011) Online distance metric learning for object tracking. IEEE Trans Circ Syst Vid 21(12):1810–1821
24. Wang F, Yu S, Yang J (2008) A novel fragments-based tracking algorithm using mean shift. In: International conference on control, automation, robotics and vision (ICARCV), pp 694–698
25. Wang XG, Feng B, Bai X et al (2014) Bag of contour fragments for robust shape classification. Pattern Recogn 47(6):2116–2125

26. Yang F, Lu HC, Zhang WL et al (2012) Visual tracking via bag of features. IET Image Process 6(2):115–128
27. Yang F, Lu HC, Yang M-H (2014) Robust superpixel tracking. IEEE Trans Image Process 23(4):1639–1651
28. Zhang KH, Song H (2013) Real-time tracking via online weighted multiple instance learning. Pattern Recogn 46:397–411
29. Zhang KH, Zhang L, Yang M-H (2014) Fast compressive tracking. IEEE Trans Pattern Anal Mach Intell 36(10):2002–2015

Heng Fan received his B.E. degree in College of Science, Huazhong Agricultural University (HZAU), Wuhan, China, in 2013. He is currently pursuing the M.Sc. degree in the College of Engineering, Huazhong Agricultural University (HZAU), Wuhan, China. His research interests include computer vision, pattern recognition and machine learning.

Jinhai Xiang received the M.E. degree in Computer Science from China University of Geoscience (Wuhan) in 2003, and the Ph.D. degree in Computer Architecture from Huazhong University of Science & Technology (HUST) in 2014. He is now an Associate Professor in the College of Informatics, Huazhong Agricultural university, Wuhan, China. His main interests include computer vision and machine learning.

Liang Zhao is a lecture in College of Informatics, Huazhong Agricultural University (HZAU), Wuhan, China. She received the M.E. degree in Computer Science Department from Central China Normal University in 2005, and the Ph.D. degree in College of Resources and Environment from Huazhong Agricultural University (HZAU) in 2015. Her research interests include machine learning, data mining, and wireless sensor networks.