Noname manuscript No.
(will be inserted by the editor)

Soft-assigned bag of features for object tracking

Tongwei Ren · Zhongyan Qiu · Yan Liu · Tong Yu · Jia Bei

Received: date / Accepted: date

Abstract Hard assignment based bag of features (BoF) representation inevitably brings in quantization errors, which may lead to inaccuracy even failure in object tracking. In this paper, we propose a novel soft-assigned bag of features tracking approach (SABoF), in which soft assignment is utilized to improve the robustness and discrimination of BoF representation. After labeling the tracked target, we first randomly sample the circle patches with adaptive size within and outside the labeled target, extract the local features from the patches, and construct the codebooks by k-means clustering. When tracking in a new frame, we generate the BoF representation of each candidate target, and select the most similar candidate target in the previous tracked result based on BoF representation. To improve tracking performance, we also continuously update the codebooks and refine the tracking results. Experiments show that our approach outperforms the state-of-the-art tracking methods under complex tracking conditions.

Keywords Soft assignment · Bag of features · Object tracking · Visual representation

Tongwei Ren
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

Zhongyan Qiu
School of Software, Nanchang University, Nanchang, China

Yan Liu
Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
Tel.: +852-2766-7241
Fax: +852-2766-7241
E-mail: csyliu@comp.polyu.edu.hk

Tong Yu
Software Institute, Nanjing University, Nanjing, China

Jia Bei
Software Institute, Nanjing University, Nanjing, China

1 Introduction

Visual object tracking, as one of the most important topics in multimedia and computer vision, aims to estimate the target states in video sequences [1]. It draws much research attention in the past decades and plays a critical role in various applications, such as video summarization [2], event surveillance [3], human-computer interaction [4], video annotation [5] and robot control [6]. Current object tracking technology can provide good tracking results in the controlled conditions, but it is still challenging to accurately and robustly track objects in complex conditions, such as partial occlusion, illumination variation and similar background [7].

One of the essential problems in object tracking is what kind of visual representation is used to describe the spatio-temporal characteristics of target appearance [8]. In one respect, a good visual representation should be robust enough to suit the various changes of target appearance in different complex conditions. Meanwhile, it should be discriminative to the candidate targets with similar appearances to accurately detect target position [9].

According to the difference of feature types, visual representation for object tracking can be roughly categorized into global representation and local representation. Global representation extracts the visual features from the whole target to describe its global characteristics, such as color, texture and shape. It usually requires low computation cost, and provides good tracking results in the controlled conditions. It can also handle the slight illumination variation and target deformation with statistic features. But when dealing with the challenging conditions, such as partial occlusion and illumination variation, global representation usually has bad performance. Local representation focuses on the local information of the target, and represents the target with a collection of local features. For local information is easy to keep consistent in different conditions, local representation can handle illumination variation and target deformation. Meanwhile, for the local information of different parts do not influence each other, local representation is robust and discriminative in the conditions of partial occlusion and similar background. But compared to global representation, local representation requires much higher computational cost for feature extraction and similarity measurement. Recently, bag of features (BoF) is proposed to represent a collection of local features with a singular feature by assigning local features to the predefined codewords and counting the occurrences of these codewords. For its simplicity and robustness, bag of features has been widely used in many applications, including image classification [10], object categorization [11], and information retrieval [12]. And it is first applied in object tracking by Yang *et al.* in [13,14].

BoF tracking obtains good performance in some complex conditions, but its hard assignment strategy inevitably brings in quantization errors, which may cause the inaccuracy in similarity measurement and even result in failures in object tracking. Fig. 1(a)-(c) illustrate the drawback of hard assignment based BoF representation. For quantization errors, the assigned codewords collection of the correct candidate target in frame F^t (indicated with orange rectangle)

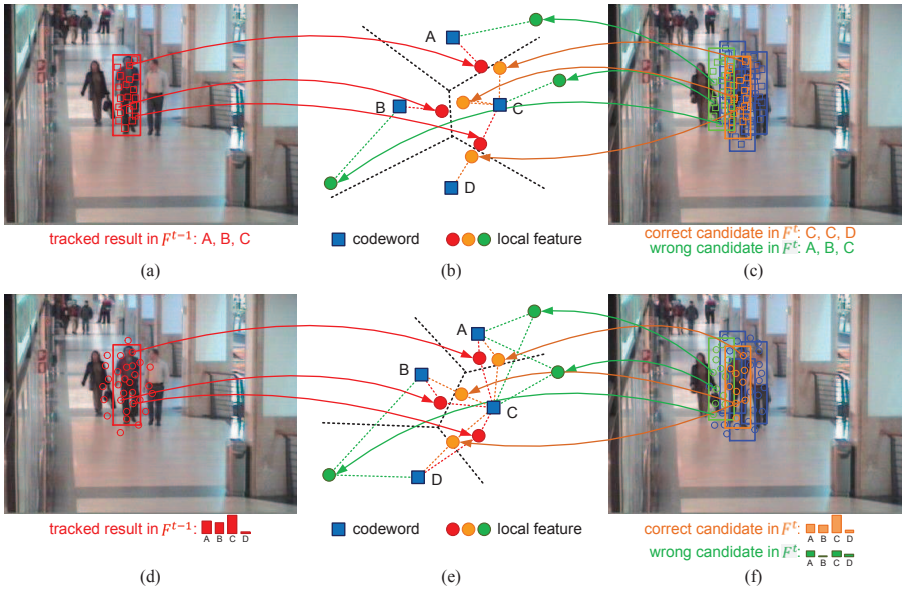

Fig. 1 Comparison of our approach and hard-assigned BoF tracking in target representation. (a) and (d) Previous frame F^{t-1}. (b) Patch assignment result of the traced result and the candidate targets by hard-assigned BoF tracking. (c) and (f) Current frame F^t. (e) Patch assignment result of the traced result and the candidate targets by soft-assigned BoF tracking.

is less similar than a wrong candidate target (indicated with green rectangle) to the tracked result in frame F^{t-1} (indicated with red rectangle). It means that the wrong candidate target will be selected as the tracked result in frame F^t, which leads to inaccuracy even failure in tracking.

In order to tackle the aforementioned problem, we propose a novel soft-assigned bag of features tracking approach (SABoF), which utilizes soft assignment strategy [15] to assign local features to the codewords. Instead of hard assigning each local feature to one codeword, our approach assigns each local feature to several nearest codewords with different weights, and determines the assigned weights according to the similarities between local features and their assigned codewords. Soft assignment strategy makes bag of features representation more robust to the changes of object appearance, and more discriminative to the object and similar background. Fig. 1(d)-(f) show the patch assignment result of our approach. To better describe the target and background, we sample the patches within and outside the target, and use them together in codebook construction. It changes the distribution of codewords, and increases the discrimination of our approach. Meanwhile, for each local feature is assigned to several nearest codewords, two near patches are easily assigned to the same codewords with similar assigned weights, even their nearest codewords are different. To two dissimilar patches, their assigned weights are usually different even the patches are assigned to the

same codewords. It makes the correct candidate target in frame F^t more similar to the tracked result in frame F^{t-1} than the wrong candidate target in BoF representation. Some preliminary results of our approach were presented in [42]. In this paper, we further analyze the advantages of soft assignment strategy in improving the robustness and discrimination for BoF tracking. We also validate the performance of our approach on the challenging video sequences which are widely used in tracking result evaluation in the previous works, and compare our approach with seven dominant tracking methods.

The main contribution of our approach includes: First, we propose an effective and robust tracking approach based on soft-assigned BoF representation, which obtains better performance than the state-of-the-art tracking methods on the challenging video sequence. Though BoF representation with soft assignment has been widely used in object categorization and other applications, to the best of our knowledge, it is the first time to use it in object tracking. Second, we analyze the advantages of soft assignment in improving the robustness and discrimination of BoF representation in tracking. Experiments show that soft assignment can effectively increase the accuracy of BoF tracking on all the videos and avoid failure under some complex conditions. Finally, we improve the efficiency and effectiveness of BoF representation generation according to the characteristic of object tracking. Specifically, we extend the patch sampling range to increase the discrimination of BoF representation, change patch shape to share them in representing different candidate targets, and use adaptive patch size to adapt the continuous changes of target size.

The remaining part of the paper is organized as follows. In Section 2, we briefly review the most related work in visual representation for tracking, bag of features and soft assignment. Then, we introduce soft-assigned bag of features, and analyze its advantages in object tracking by comparing to hard-assigned bag of features in Section 3. In Section 4, we describe the details of our tracking approach using soft-assigned bag of features. Experiment results are shown and discussed in Section 5. Finally, we conclude the paper with future work.

2 Related work

2.1 Visual representation for tracking

Visual representation is one of the essential problems in object tracking. There are many issues leading to the difficulty to provide robust and discriminative visual representation, including various environment, object appearance deformation, and low camera quality [8]. According to the difference of feature types, visual representation in object tracking can be roughly categorized into global visual representation and local visual representation.

Global visual representation extracts the features from the whole target to describe the global characteristics of target appearance. Pérez *et al.* propose a Monte Carlo tracking method using color histogram within a probabilistic framework [16]. Allili *et al.* utilize level set active contour representation to segment the object from background and solve the problem of non-rigid object tracking [17]. Santner *et al.* use constant-brightness-constraint optical-flow in mean-shift tracker to improve the stabilization in tracking [18]. Hu *et al.* use the pixel matrix in subspace learning to model the target and apply it in foreground segmentation and tracking [19]. Global visual representation is usually simple and efficient in object tracking, but it is sensitive to target appearance changes caused by partial occlusion, illumination variation or other complex conditions.

Local visual representation extracts the features from the local regions to model the local structure of target appearance. Kim represent video content with corner feature, and generate a set of corner point trajectories by dynamic multi-level grouping [20]. Zhou *et al.* use SIFT to represent the interest points in the target, and combine it with mean-shift based similarity of color histogram in tracking [21]. He *et al.* apply SURF descriptor in object tracking, and evaluate its performance in appearance change [22]. Wang *et al.* construct the perspective of mid-level vision with structural information captured from the superpixels to handle the large changes of target scale, motion and deformation [23]. Local visual appearance captures the local structure of the tracked object. It can handle the appearance changes caused by partial occlusion, illumination variation or target deformation. But it requires high computational cost and sometimes suffers from disturbance.

2.2 Bag of features

Bag of features provides a simple and powerful representation of visual content with a collection of local features. It evolved from bag of words in document processing and texton methods in texture analysis [24]. The basic idea of bag of features is to sample the patches from visual content, extract the features from each patch and quantify the features according to codewords, and represent the image content with the codeword distribution.

There are several problems in bag of features representation, such as how to sample the patches, how to describe the patches with features, and how to characterize the distribution of the codewords. To patch sampling strategy, Nowak *et al.* evaluate the representative sampling strategies, and find that the number of sampled patches plays a more important role than sampling strategy in classification [25]. To feature descriptor, Lowe proposes SIFT descriptor to provide scale and rotation invariant description of image local information [26], which is further improved for compact description and large scale applications [27]. Some color and texture descriptors for global information description [28] is also used to describe the sampled patches for their efficiency in extraction

and similarity measurement. To codeword distribution characterization, a sparse vector of codeword occurrence frequency is usually used.

For its simplicity and good performance, bag of features representation is utilized in many applications. Li *et al.* represent each natural image with a collection of codewords obtained by unsupervised learning, and propose a bayesian hierarchical model to recognize scene categories [29]. Jiang *et al.* optimize various factors in BoF representation, such as detector choices and vocabulary size, and apply it in object categorization and video retrieval [11]. Wang *et al* use locality-constrained linear coding in spatial pyramid matching and apply linear classifier in image classification [10]. Jégou *et al.* improve BoF representation with Hamming embedding and weak geometric consistency constraints, and apply it in large scale image search [12]. Yang *et al.* use bag of features for target representation in objecting tracking, and combine it with incremental PCA tracking [13,14].

2.3 Soft assignment

Soft assignment, also named *soft weighting*, is a widely used weighting scheme for BoF representation [30]. In the conventional weighting schemes, each keypoint or patch is only assigned to its nearest visual word. For visual words are usually generated by feature clustering or random sampling [31], the similar keypoints or patches may be assigned to the different visual words. To overcome the drawback, soft assignment strategy assigns each keypoint or patch to its several nearest visual words, and represents it with a weight vector to these visual words.

Jiang *et al.* first propose a soft-weighting scheme to improve BoF representation performance, in which the weight of each visual word in an image is calculated as the sum of its similarities to the keypoints treating it as their top-N nearest visual words [11]. Philbin *et al.* assign each patch to several visual words nearby in the descriptor space and show its benefit for retrieval with large vocabularies [15]. Gemert *et al.* allow some ambiguity in assigning codewords to image features, which used to solve the problem of codeword uncertainty and plausibility, and improve the performance of scene categorization [32]. Zhu *et al.* map multiple visual words to each keypoint with soft weighting, and the generated BoF representation for learning visual concept classifiers [33].

3 Soft-assigned bag of features

3.1 Hard assignment for BoF representation

In the traditional bag of features, each local feature $\boldsymbol{f_i}$ is simply assigned to its nearest codeword $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$, and represented by once occurrence of this codeword:

$$\hat{\boldsymbol{c}}_{\boldsymbol{i}} = \arg\min_{\boldsymbol{c_k}} \|\boldsymbol{f_i} - \boldsymbol{c_k}\|, k \in \{1, ..., N_C\} \tag{1}$$

where $\boldsymbol{c_k}$ is the kth codeword and N_C is the number of codewords.

Such hard assignment strategy inevitably brings in quantization errors in generating BoF representation. It may only cause slight influence in some applications, such as image retrieval and classification, but lead to inaccuracy even failure in object tracking. There are several reasons: First, compared to the applications such as image retrieval and classification, the local features are extracted from the very limited target regions in tracking. It causes the distances between different codewords to be much smaller, and easily leads to assigning the near local features with various changes to different codewords in partial occlusion, illumination variation or other complex conditions. Second, the candidate targets for feature extraction in object tracking are usually sampled around the target position in the previous frame. They are sometimes similar in appearance, for example, the object is passing another object or background with similar color and texture. If only assign each local feature to one codeword, it discards the relationships between the local feature and other codewords, which represents the differences among the candidate targets, and may cause inaccurate tracking results. Furthermore, the local features extracted from noises will be also assigned to their nearest codewords in hard assignment, no matter how dissimilar they are, and treated completely same to other local features near to these codewords. For the number of local features to represent each candidate target in tracking is much smaller than other applications, the local features extracted from noises play obvious roles in similarity measurement and even seriously influence the tracking results.

To solve the problem, hard assignment is improved by representing each local feature with its assigned weight to the nearest codeword instead of codeword occurrence times [13]:

$$\omega(\boldsymbol{f_i}, \hat{\boldsymbol{c}}_{\boldsymbol{i}}) = \max\left(1 - \|\boldsymbol{f_i} - \boldsymbol{c_k}\|\right), k \in \{1, ..., N_C\} \tag{2}$$

where $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ is the codeword which $\boldsymbol{f_i}$ is assigned to according to Equation (1).

Though weighted hard assignment quantitatively describes the relationship between local feature and its assigned codeword, it cannot solve the above problems. In one respect, since each local feature is still assigned to one codeword, the distance between two near local features keeps large if they are assigned to different codewords. In the other respect, though the similarities between local features and their assigned codewords are quantitatively described by assigned weights, two different local features cannot be distinguished if they are assigned to a codeword with the same assigned weights. Above all, weighted hard assignment is not robust or discriminative enough for object tracking.

3.2 Soft assignment for BoF representation

Different to hard assignment, soft assignment strategy assigns each local feature to several nearest codewords and represents it with a collection of

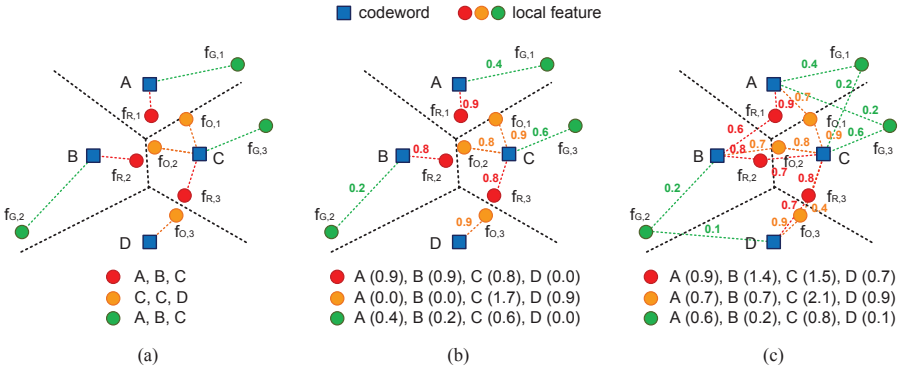

Fig. 2 Example of robustness comparison. (a) Result of hard assignment. (b) Result of weighted hard assignment. (c) Result of soft assignment.

its assigned weights to these codewords:

$$\boldsymbol{b}_{\boldsymbol{f_i}} = \left\{\omega(\boldsymbol{f_i}, \hat{\boldsymbol{c}}_{\boldsymbol{i,1}}), \omega(\boldsymbol{f_i}, \hat{\boldsymbol{c}}_{\boldsymbol{i,2}}), \ldots, \omega(\boldsymbol{f_i}, \hat{\boldsymbol{c}}_{\boldsymbol{i,N_A}})\right\}, N_A \leq N_C \tag{3}$$

where $\boldsymbol{b}_{\boldsymbol{f_i}}$ is the BoF representation of $\boldsymbol{f_i}$, $\hat{\boldsymbol{c}}_{\boldsymbol{i,1}}, \hat{\boldsymbol{c}}_{\boldsymbol{i,2}}, \ldots, \hat{\boldsymbol{c}}_{\boldsymbol{i,N_A}}$ are the N_A-nearest codewords that $\boldsymbol{f_i}$ are assigned to, and $\omega(,)$ are the assigned weights of $\boldsymbol{f_i}$ to its assigned codewords.

Soft assignment strategy can improve robustness of BoF representation in object tracking. For each local feature is assigned to several codewords and represented by the assigned weights to these codewords, two near local features are easily assigned to the same codewords and the assigned weights are close, though their most nearest codewords are different. Fig. 2 shows an example of robustness comparison with different assignment strategies in BoF representation, in which the local features extracted from the tracked result $\hat{T}^{k-1}$ in the previous frame F^{k-1}, the correct candidate target T^k_{cor} in the current frame F^k and a wrong candidate target T^k_{wro} in the current frame F^t are indicated with red, orange and green circles, respectively. Fig. 2(a) shows the result of hard assignment. The local features from $\hat{T}^{k-1}$ are assigned to codeword A, B and C respectively, and the local features from T^k_{cor} are assigned to codeword C, C and D for quantization errors. Meanwhile, the local features from T^k_{wro} are assigned to codeword A, B and C, though they are farther to the local features from $\hat{T}^{k-1}$. Especially, the local feature $f_{G,2}$ in the bottom-left corner is far away from all the codewords, but it is still assigned to codeword B. In this way, the BoF representation of T^k_{wro} is same to $\hat{T}^{k-1}$, and T^k_{wro} will be selected as the tracked result in frame F^k. Fig. 2(b) shows the result of weighted hard assignment. With the assigned weight of local feature to its assigned codeword, the BoF representation similarity of T^k_{wro} and $\hat{T}^{k-1}$ is decreased from 1 to 0.56. But it is higher than the BoF representation similarity of T^k_{cor} to $\hat{T}^{k-1}$, which is only 0.10 for their near local features are assigned to different codewords. It means T^k_{wro} will still be selected as the tracked result in frame F^k. Fig. 2(c) shows the result of soft assignment.

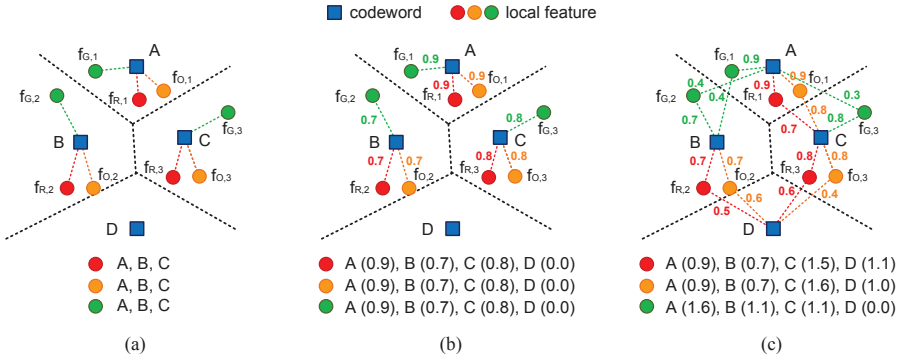

Fig. 3 Example of discrimination comparison. (a) Result of hard assignment. (b) Result of weighted hard assignment. (c) Result of soft assignment.

For each local feature is assigned to several nearest codewords (two nearest codewords here), the near local features from T_{cor}^k and $\hat{T}^{k-1}$ are assigned to the approximate codewords, and the similarity of their BoF representations increases to 0.52. And for the local features from T_{wro}^k and $\hat{T}^{k-1}$ are much farther, the similarity of their BoF representations is only 0.21. It results in the correct selection of T_{cor}^k as the tracked result in F^k, and the accuracy of object tracking is improved.

Another advantage of soft assignment is improving the discrimination of BoF representation in object tracking. For each local feature should be assigned to more than one codewords and represented by the assigned weights to these codewords, two different local features can be distinguished even they have the same nearest codeword with the same assigned weights. Fig. 3 shows an example of discrimination comparison with different assignment strategies in BoF representation, in which the local features indicated with red, orange and green squares have the same meaning as in Fig. 2. Similarly, Fig. 3(a) shows the result of hard assignment. The local features extracted from T_{cor}^k and T_{wro}^k are assigned to the same codewords as $\hat{T}^{k-1}$. It means T_{cor}^k and T_{wro}^k have the same BoF representation similarities to $\hat{T}^{k-1}$ and they cannot be distinguished. Fig. 3(b) shows the result of weighted hard assignment. Though the local features extracted from T_{cor}^k are nearer to $\hat{T}^{k-1}$ than T_{wro}^k, they still cannot be distinguished for they have the same assigned weights to the codewords. Fig. 3(c) shows the result of soft assignment. For the local features extracted from T_{cor}^k are nearer to $\hat{T}^{k-1}$ than T_{wro}^k, they are assigned to the same codewords with the close assigned weights, and their BoF representation similarity is 0.93, which is much higher than the similarity of BoF representations between T_{wro}^k and $\hat{T}^{k-1}$. It shows that soft assignment effectively increases the discrimination of BoF representation and improves the accuracy of object tracking.

4 Soft-assigned BoF tracking

We first initialize object tracking by manually labeling the tracked target in the first frame, and automatically labeling the tracked results in the following several frames to collect sufficient local features for codebook construction. Then, we extract the local features from the shared patches with adaptive size, and construct the initial codebooks based on the patches sampled within and outside the tracked results. When tracking in a new frame, we locate the target position by selecting the most similar candidate target to the tracked result in the previous frame based on soft-assigned BoF representation. To improve tracking performance, we continuously update the codebooks and refine the tracking result with incremental PCA tracking.

4.1 Initialization

Similar to other tracking methods, we initialize object tracking by manually labeling the tracked target in the first frame with a rectangle. With the manual labeling, we can obtain the initial target state, including center coordinate $(\hat{x}^0_{cen}, \hat{y}^0_{cen})$, width $\hat{w}^0$, height $\hat{h}^0$ and rotation angle $\hat{\varphi}^0$. Meanwhile, it requires more labeled frames to obtain sufficient local features to generate the codebooks, which can be obtained by manually labeling or other tracking methods. As BoF tracking, we use incremental PCA tracking [13] to label the first several frames in our approach.

4.2 Local feature extraction and codebook generation

Assume N_R frames are initially labeled, we randomly sample some patches in each frame. To make our approach more effective and efficient, we improve the patch sampling in BoF tracking [14]: First, besides sampling N_P patches within the target, we also sample N_P patches from the region around but outside the target, and use the sampled patches together in codebook construction. The additional patches outside the target change the distribution of the codewords. It makes the local features extracted from the target and background easier to be assigned to different codewords, which is helpful to increase the discrimination of BoF representation. Second, we use circle patches instead of square patches, and share the sampled patches among the candidate targets. It efficiently reduces the computation cost in candidate target selection. More details will be introduced in Sec. 4.3. Finally, we utilize the adaptive patch size to adapt to continuous object size change, which is difficult to solve in BoF tracking [14]. To determine patch size, we predict the target width $\tilde{w}^t$ and height $\tilde{h}^t$ in frame F^t with N_R previous frames:

$$\begin{aligned} \tilde{w}^t &= \hat{w}^{t-1} + (\hat{w}^{t-1} - \hat{w}^{t-N_R}) \times \frac{N_R}{N_R - 1}, \\ \tilde{h}^t &= \hat{h}^{t-1} + (\hat{h}^{t-1} - \hat{h}^{t-N_R}) \times \frac{N_R}{N_R - 1}, \end{aligned} \tag{4}$$

where $\hat{w}^{t-1}$ and $\hat{h}^{t-1}$ are the target width and height in frame F^{t-1}, and $\hat{w}^{t-N_R}$ and $\hat{h}^{t-N_R}$ are the target width and height in frame F^{t-N_R}, respectively. And the patch radius r^t in frame F^t is determined as:

$$r^t = \sqrt{\frac{\tilde{w}^t \tilde{h}^t}{\pi N_P}}, \tag{5}$$

where N_P is the number of patches sampled within or outside the target as above.

Then, we extract the features from the sampled patches and represent each patch p_i^t in frame F^t with a feature set $\{\boldsymbol{f_{1,i}^t}, \boldsymbol{f_{2,i}^t}, \ldots, \boldsymbol{f_{N_f,i}^t}\}$, here $\boldsymbol{f_{k,i}^t}$ is the kth feature of p_i^t and N_f is the number of feature types. To each type of feature, we cluster the $2 \times N_P \times N_R$ features from the N_R initially labeled frames into N_C clusters by k-means algorithm [34]. The cluster centers are treated as the codewords to compose the initial codebook, and each codeword $\boldsymbol{c_{k,*}}$ is represented with a kth type feature. In this way, we generate N_f codebooks, and the tracked target can be represented by bag of features with these codebooks.

To each local feature, we use an exponential distance function instead of Euclidean distance in Equation (2) to calculate its assigned weights to the assigned codewords, and set its assigned weights to other codewords to 0:

$$\omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,j}}) = \begin{cases} \exp\left(-\frac{\|\boldsymbol{f_{k,i}^t} - \boldsymbol{c_{k,j}}\|^2}{\sigma^2}\right), & \boldsymbol{f_{k,i}^t} \text{ is assigned to } \boldsymbol{c_{k,j}} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where σ is a parameter to adjust the weight values, which is usually influenced by the distances between codewords. So a local feature $\boldsymbol{f_{k,i}^t}$ can be represented as a vector of its assigned weights to all the N_C codewords on the kth feature:

$$\boldsymbol{\omega_{f_{k,i}^t}} = \left[\omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,1}}), \omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,2}}), \ldots, \omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,N_C}})\right]. \tag{7}$$

Thus, the BoF representation of the whole target T^t on the kth feature is represented as a histogram of the assigned weight sums of all the N_P features:

$$\boldsymbol{h_{T^t,k}} = \left[\sum_{i=1}^{N_P} \omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,1}}), \sum_{i=1}^{N_P} \omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,2}}), \ldots, \sum_{i=1}^{N_P} \omega(\boldsymbol{f_{k,i}^t}, \boldsymbol{c_{k,N_C}})\right]. \tag{8}$$

In this way, the target can be represented with BoF representation using N_f histograms of assigned weight sums on different codebooks.

4.3 Candidate target selection

When tracking the target in a new frame F^t, we sample N_T candidate targets around the center position $(\hat{x}_{cen}^{t-1}, \hat{y}_{cen}^{t-1})$ of the tracked result $\hat{T}^{t-1}$ in the previous frame F^{t-1}, and generate the BoF representation for each candidate target. Local feature extraction and assignment in BoF representation generation are

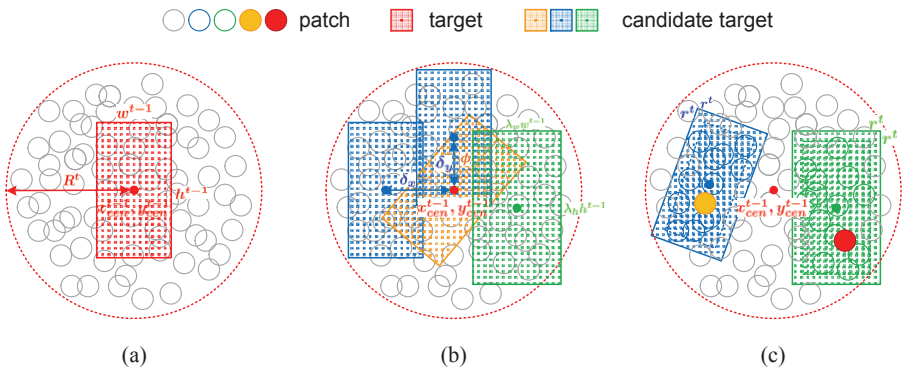

Fig. 4 Example of candidate target selection. (a) Sample patches around the center position $(\hat{x}_{cen}^{t-1}, \hat{y}_{cen}^{t-1})$ of $\hat{T}^{t-1}$. (b) Sample candidate targets around the center position $(\hat{x}_{cen}^{t-1}, \hat{y}_{cen}^{t-1})$ of $\hat{T}^{t-1}$. (c) Adjust patch coverage of candidate targets by adding or ignoring the patches.

time consuming. If independently sample the patches within each candidate target as BoF tracking, it requires $N_P \times N_T$ times feature extraction and $N_P \times N_C \times N_f \times N_T$ times feature assignment, here N_C is the codeword number in each codebook and N_f is the codebook number as above.

Considering the candidate targets have much overlapping region, we share the patches among different candidate targets to reduce computational cost. We first sample several patches around the center position $(\hat{x}_{cen}^{t-1}, \hat{y}_{cen}^{t-1})$ of $\hat{T}^{t-1}$ within the distance R^t (Fig. 4(a)), here R^t is determined by the predefined affine parameters to select the candidate targets. Assume the center shift parameters are δ_x and δ_y in x and y coordinates, and the scale parameters are λ_w and λ_h in width and height (Fig. 4(b)), the distance R^t is roughly calculated by ignoring the rotation parameter ϕ:

$$R^t = \max\left\{\delta_x + \frac{\lambda_w \hat{w}^{t-1}}{2}, \delta_y + \frac{\lambda_h \hat{h}^{t-1}}{2}\right\}. \tag{9}$$

Within the circle centered in $(\hat{x}_{cen}^{t-1}, \hat{y}_{cen}^{t-1})$ with the radius of R^t, we randomly sample $\tilde{N}_P$ patches and assign their features to the codewords. Sampled patch number $\tilde{N}_P$ should be appropriate. Large $\tilde{N}_P$ value helps candidate targets to cover sufficient patches, but increases computational cost; and small $\tilde{N}_P$ value reduces computational cost, but may not provide sufficient patches to each candidate target. We calculate $\tilde{N}_P$ by setting the sampling region with the same patch density to the requirement of each candidate target:

$$\frac{\tilde{N}_P}{\pi (R^t)^2} = \frac{N_P}{\tilde{w}^t \tilde{h}^t}. \tag{10}$$

Based on Equation (5) and (10), $\tilde{N}_P$ can be calculated as:

$$\tilde{N}_P = \left(\frac{R^t}{r^t}\right)^2. \tag{11}$$

After extract the features from each patch and assign them to the codewords, we generate the BoF representations of the candidate targets. As shown in Fig. 4(c), the patches covered by each candidate target can be rapidly detected. Assume the width and height of the candidate target T_l^t are w_l^t and h_l^t, a patch is covered by T_l^t if its center locates within the $(w_l^t - 2r^t) \times (h_l^t - 2r^t)$ size region with the same center position of T_l^t (indicated with dashed rectangles). If a candidate target covers less than N_P patches, we randomly sample the additional required patches within it, for example, the orange patch in the blue candidate target in Fig. 4(c), and reuse them in BoF representation of other candidate targets. If a candidate target covers more than N_P patches, we randomly ignore several patches in BoF representation generation, for example, the red patch in the green candidate target in Fig. 4(c). For the previous sampling keeps patch density, the number of added and ignored patches is usually not large. It means that the total number of patches requiring feature extraction and assignment is much smaller than BoF tracking, and the computational cost is obviously reduced.

Based on Equation (8), we obtain the BoF representation of each candidate target T_l^t with N_f histograms $\{\boldsymbol{h}_{\boldsymbol{T_l^t,1}}, \boldsymbol{h}_{\boldsymbol{T_l^t,2}}, \ldots, \boldsymbol{h}_{\boldsymbol{T_l^t,N_f}}\}$. We measure similarities between the candidate targets in frame F^t and the tracked result $\hat{T}^{t-1}$ in frame F^{t-1} on BoF representation, and select the most similar candidate target as the tracked result $\hat{T}^t$ in frame F^t:

$$\hat{T}^t = \arg\min_{T_l^t} \sum_{k=1}^{N_f} \|\boldsymbol{h}_{\boldsymbol{\hat{T}^{t-1},k}} - \boldsymbol{h}_{\boldsymbol{T_l^t,k}}\|, l \in \{1, ..., N_T\}. \tag{12}$$

4.4 Codebook update and result refinement

For the appearance of tracked object continuously changes in video sequence, we should update the codebooks to generate BoF representation. Thus, after obtaining the target position in each frame, we add the N_P patches within the tracked result and N_P patches from other candidate targets but outside the target, and remove the oldest $2 \times N_P$ patches in codebook generation. Meanwhile, k-means clustering required in codebook update is time-consuming. To reduce computational cost, we simply calculate the new means of each cluster and regenerate the BoF representation after processing a frame, and completely update the codebooks by approximate k-means clustering per N_R frames [34].

For BoF representation ignores the global information of the target, it is helpful to adopt some global feature based tracking methods to refine the tracking results [14]. As BoF tracking, we adopt incremental PCA tracking to refine the affine parameters, including center shift, weight and height scale, and rotation angle. When the similarity between the tracked result in the previous frame and the most similar candidate target in the current frame is smaller than a predefined threshold, the affine parameters are adjusted by combining the parameters of incremental PCA tracking with a weight α. Note here,

same to BoF tracking, result refinement only works when all the candidate targets are seriously dissimilar to the tracked result in the previous frame, which seldom happens in tracking procedure, and it never directly changes the tracking results in our approach.

5 Experiments

To validate our approach, we implement it with MATLAB, and carry out the experiments on a computer with Dual Core 2.70GHz CPU and 4GB main memory. After introducing the dataset and parameter setting used in experiment, we first show the qualitative tracking results of our approach under several complex condition. Then, we compare our approach with seven state-of-the-art tracking methods with quantitative evaluation on twelve challenging video sequences. Finally, we analyze the limitations of our approach in discussion.

5.1 Parameter selection

Several key parameters in our approach influences tracking performance. To demonstrate the advantage of soft assignment, we use similar parameter setting to original BoF tracking [14]. We select $N_T = 300$ candidate targets in each frame. Considering the local features outside the target are also used in clustering, we reduce the patch number to $N_P = 40$ within each candidate target. We use 32-bin HSV color histogram [28] (only 4 bins for value to reduce the influence of illumination variation) and 59-bin local binary pattern texture descriptor [35], and generate two codebooks which are composed of constantly 20 codewords and updated per $N_R = 5$ frames. The combination weight α in result refinement equals 0.7 to assign incremental PCA tracking more importance in refinement. In soft assignment, each patch is assigned to the nearest $N_A = 3$ codewords, and σ in assinged weight calculation is set to 1/9, for the distances between codewords are small in tracked object representation.

5.2 Tracking results

We evaluate our approach in five typical complex conditions, including partial occlusion, illumination variation, similar background and object encounter. All the experiments in this subsection are carried out on CAVIAR dataset [43]. CAVIAR dataset aims to evaluate the performance of local features in the tasks of human behavior detection and recognition. It consists of 80 video clips about human behaviors in different scenarios of interest, including walking alone, meeting with others, entering and exiting shops and leaving a package. In each video sequence, one to twenty targets are manually annotated with the rectangles to represent their states, including center position, weight, height

and rotation. The manual annotation is treated as the ground truth in our experiments.

Partial occlusion: Occlusion is a general yet crucial problems in object tracking. Partial occlusion may change the target appearance and enlarge the difference between the targets in successive frames. Fig. 5 shows an example of our tracked result in partial occlusion condition on video sequence *WalkByShop1front*. Yellow dashed rectangle and red solid rectangle indicate ground truth and our tracked result respectively, and frame number is labeled in top-left corner of each frame. The same indication is used in Fig. 6-Fig. 9. In this example, the tracked woman is always partially occluded by a man from her appearance to disappearance. Our approach keeps the accuracy of the target center position in tracking and only has small scale difference to the ground truth. It shows that our approach can handle serious partial occlusion.

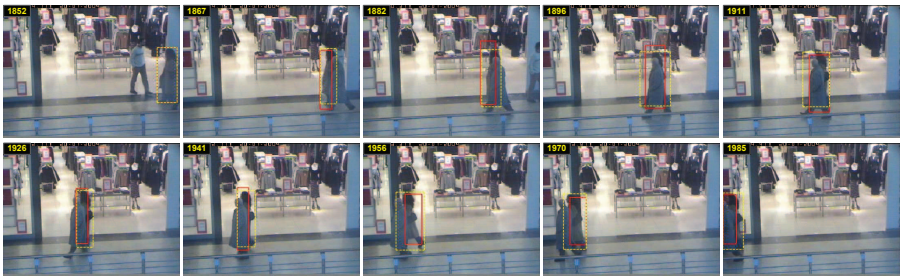

Fig. 5 Example of our tracking result in partial occlusion condition on video sequence *WalkByShop1front*.

Illumination variation: Various illumination is also a challenging problem in object tracking. For environment change or object motion, the illumination on target may continuously vary. Fig. 6 shows an example of our tracked result in illumination variation condition on video sequence *Browse_WhileWaiting1*. In this example, the tracked man walks from the shaded region to the sunny region and back to the shaded region. The illumination variation on the man seriously changes the target appearance in the procedure. Our approach has some scale and rotation differences to the ground truth, but keeps the accuracy of the target center position in tracking. It shows that our approach can handle violent illumination variation well.

Target deformation: Non-rigid targets are easily deformed in tracking. It may cause the appearance change and even self-occlusion of the targets. Fig. 7 shows an example of our tracked result in target deformation condition on video sequence *Browse3*. In this example, the tracked man raises his arms and put them down, and walks toward the bottom-right corner with smaller and smaller appearance. Though our approach does not cover the arms of the man in some frames and has some rotation difference to the ground truth, it follows the target center position in all the procedure. It shows that our approach can handle various target deformation.

Fig. 6 Example of our tracking result in illumination variation condition on video sequence *Browse_WhileWaiting1*.

Fig. 7 Example of our tracking result in target deformation condition on video sequence *Browse3*.

Similar background: Object tracking can be considered as a classification problem to classify video content into target and background. When the target passes the background with similar appearance, object tracking will become difficult. Fig. 8 shows an example of our tracked result in similar background on video sequence *EnterExitCrossingPaths2cor*. In this example, the tracked man passes the wall and the hallway with similar color to his coat. Our approach obtains a similar result to the ground truth with some slight shift of the center position. It shows that our approach can handle similar background.

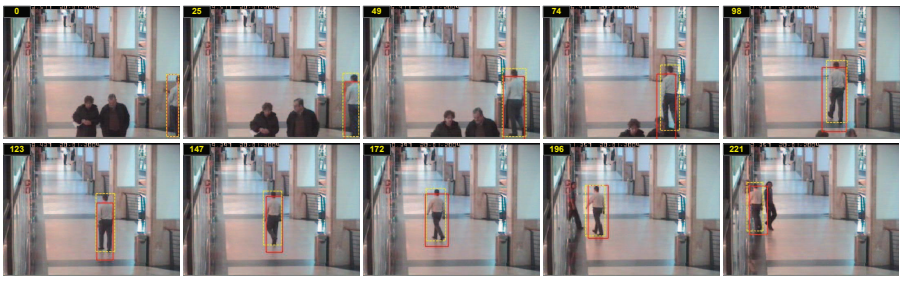

Fig. 8 Example of our tracking result in similar background condition on video sequence *EnterExitCrossingPaths2cor*.

Object encounter: Object encounter causes several problems in object tracking, such as occlusion. Especially when the encountered objects have similar appearances, it is very difficult to distinguish them. Fig. 9 shows an example of our result in object encounter on video sequence *TwoEnterShop2front*. In this example, the tracked man meets some persons with similar color coats, and partially occluded with them. Our approach obtains a similar result to the ground truth and only has some small scale difference to the ground truth. It shows that our approach can handle object encounter well.

Fig. 9 Example of our tracking result in object encounter condition on video sequence *TwoEnterShop2front*.

5.3 Quantitative comparison

To illustrate the performance of our approach, we compare it with seven state-of-the-art tracking methods on twelve challenging video sequences, including *Car4*, *Car11*, *DavidIndoor*, *DavidOutdoor*, *FaceOccu2*, *Football*, *Jumping*, *OneStopEnter1front*, *ShopAssistant2cor*, *Singer1*, *ThreePastShop2cor* and *WalkByShop1front*. Most video sequences were used in the compared tracking methods, and two new video sequences are selected from CAVIAR dataset to evaluate the tracking performance under complex conditions.

The compared tracking methods includes robust fragments-based tracking (Frag) [36], incremental PCA tracking (IVT) [37], multiple Instance Learning (MIL)[38], visual tracking decomposition (VTD) [39], sparsity-based collaborative model tracking(SCM) [1], tracking learning detection (TLD) [40] and original BoF tracking (BoF) [14]. Fig. 13 shows the tracking results of our approach and other seven methods on the twelve video sequences, in which the dark green dashed boxes indicate the manually-labeled ground truths and other color solid boxes indicate the tracked results of different methods.

Table 1 and Table 2 illustrate a comparison of our approach with other seven tracking methods in average center position error and average overlap rate on each video sequence, respectively. Here, center position error is measured as the Euclidean distance between the center positions of the tracked result and the ground truth in pixel, and overlap rate is measured by the

Table 1 Comparison of average center position errors on twelve challenging video sequences.

	Frag	IVT	MIL	VTD	SCM	TLD	BoF	SABoF
Car4	180.02	**2.89**	60.09	12.30	3.61	19.57	54.02	20.14
Car11	64.39	1.97	43.40	27.08	**1.71**	25.06	34.10	16.71
DavidIndoor	147.11	**2.76**	34.93	48.50	3.25	13.56	83.10	16.98
DavidOutdoor	89.70	52.97	37.50	62.10	67.71	172.10	20.43	**20.25**
FaceOccu2	15.56	10.25	14.17	10.42	**4.73**	18.49	13.07	6.07
Football	17.54	17.80	15.44	4.92	9.60	11.80	31.36	**3.20**
Jumping	56.51	36.85	9.96	62.99	3.79	3.94	6.97	**2.85**
OneStopEnter1front	79.51	80.90	71.56	174.77	144.25	172.15	4.29	**4.18**
ShopAssistant2cor	9.93	7.17	68.34	5.92	8.93	15.91	3.47	**1.26**
Singer1	22.19	8.51	15.18	4.17	3.83	32.65	4.82	**1.93**
ThreePastShop2cor	115.42	66.30	100.38	44.18	2.41	44.26	24.31	**1.45**
WalkByShop1front	33.79	160.12	148.11	186.65	34.43	25.31	32.22	**1.44**
Average	59.42	40.51	50.82	57.43	25.88	48.66	23.47	**6.94**

Table 2 Comparison of average overlap rates on twelve challenging video sequences.

	Frag	IVT	MIL	VTD	SCM	TLD	BoF	SABoF
Car4	0.21	0.88	0.27	0.73	**0.89**	0.63	0.43	0.77
Car11	0.11	**0.84**	0.22	0.47	0.79	0.38	0.10	0.58
DavidIndoor	0.09	0.68	0.24	0.24	**0.76**	0.56	0.13	0.43
DavidOutdoor	0.38	0.56	0.41	0.44	0.45	0.16	**0.68**	0.64
FaceOccu2	0.60	0.57	0.61	0.59	0.81	0.49	0.68	**0.86**
Football	0.58	0.55	0.57	0.80	0.66	0.61	0.63	**0.89**
Jumping	0.13	0.28	0.52	0.08	0.73	0.69	0.64	**0.86**
OneStopEnter1front	0.01	0.01	0.06	0.03	0.03	0.02	0.62	**0.82**
ShopAssistant2cor	0.79	0.67	0.64	0.72	0.64	0.55	0.77	**0.93**
Singer1	0.34	0.66	0.33	0.79	0.85	0.41	0.85	**0.95**
ThreePastShop2cor	0.14	0.15	0.14	0.16	**0.88**	0.17	0.15	0.68
WalkByShop1front	0.46	0.05	0.07	0.03	0.03	0.02	0.05	**0.95**
Average	0.32	0.49	0.34	0.42	0.63	0.39	0.48	**0.78**

intersection of the tracked result and the ground truth in area and their union [41]. Fig. 10 and Fig. 11 further present center position errors and overlap rates of eight tracking methods on each frame of each video sequence, respectively. It shows that our approach can keep the lowest average center position error and the highest average overlap rate on most video sequences while other methods fails on one or more video sequences. It means that our approach can achieve more accurate and stable tracking results than other methods under complex conditions. To the rest video sequences, such as *Car11* and *DavidIndoor*, our approach obtains worse results than the best tracking method, but it performs much better than the primary BoF tracking. It shows that soft assignment strategy plays an important role in improving the robustness and discrimination of BoF representation in object tracking.

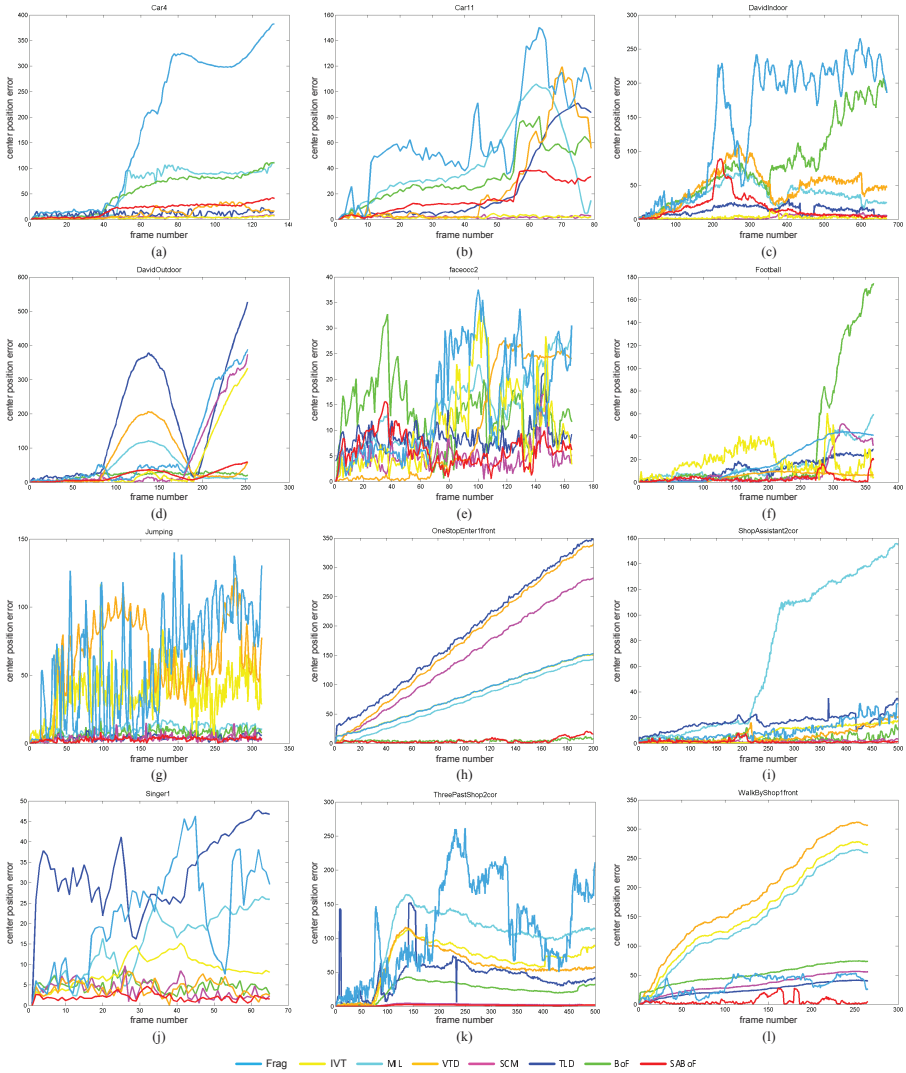

Fig. 10 Comparison of center position error on each frame. The horizontal axis is frame number, and the vertical axis is center position error. (a)-(l) Center position errors on video sequence *Car4*, *Car11*, *DavidIndoor*, *DavidOutdoor*, *FaceOccu2*, *Football*, *Jumping*, *OneStopEnter1front*, *ShopAssistant2cor*, *Singer1*, *ThreePastShop2cor* and *WalkByShop1front*, respectively.

5.4 Discussion

In experiment, we also find some limitations of our approach. For example, BoF representation only provides the local statistic information in the tracked object, but ignores the global information of the tracked target. It leads to the requirement of additional result refinement with global feature based

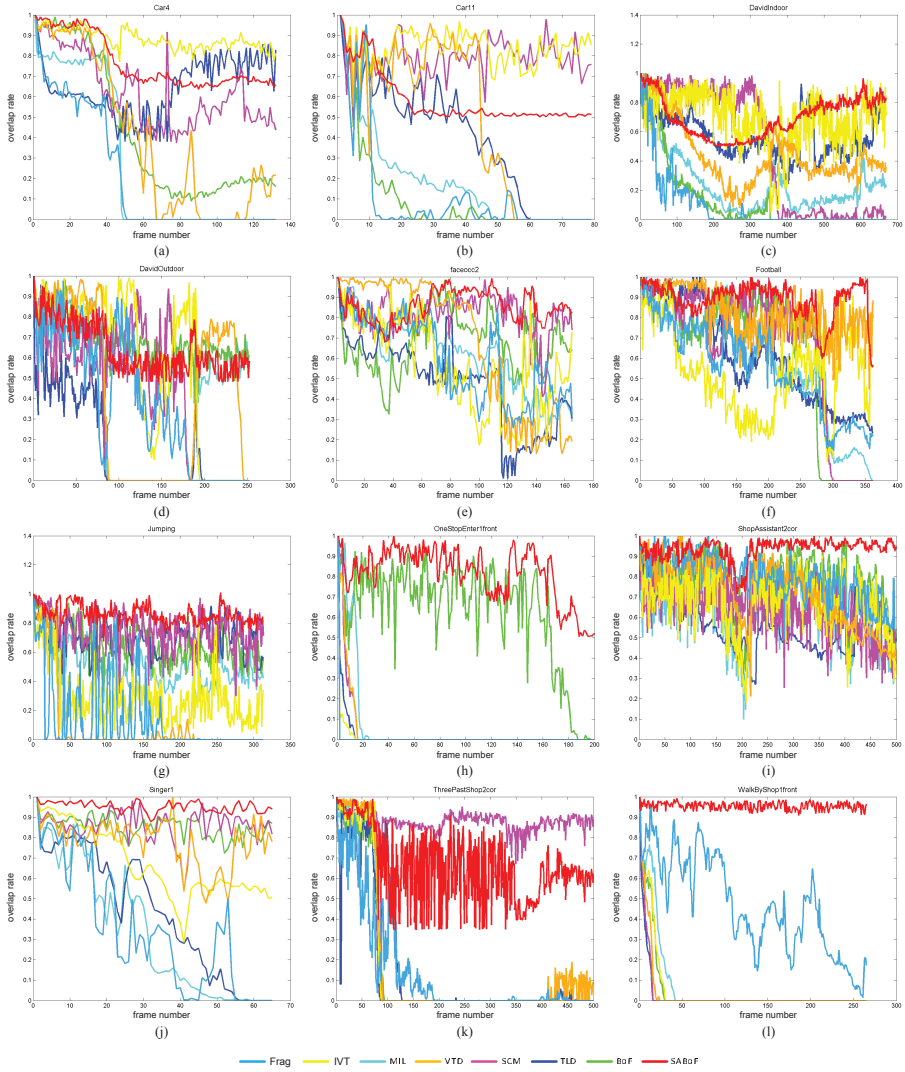

Fig. 11 Comparison of overlap rate on each frame. The horizontal axis is frame number, and the vertical axis is overlap rate. (a)-(l) Overlap rates on video sequence *Car4*, *Car11*, *DavidIndoor*, *DavidOutdoor*, *FaceOccu2*, *Football*, *Jumping*, *OneStopEnter1front*, *ShopAssistant2cor*, *Singer1*, *ThreePastShop2cor* and *WalkByShop1front*, respectively.

tracking methods, such as incremental PCA tracking. Another limitation of our approach is the dependance of codebook initialization performance. Fig. 12 shows a failure example of our approach caused by bad codebook initialization. For the person suddenly appears from totally occluded and the color of his coat is same to the surrounding region, our approach fails to track the target in the first several frames and initializes the codebooks with wrong patches. It leads to the failure of our approach in tracking.

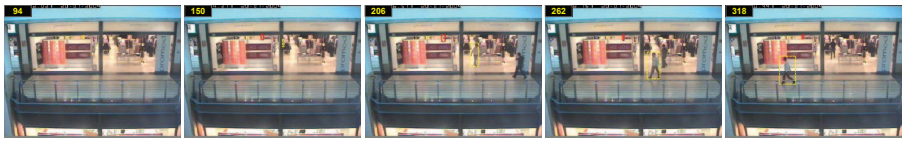

Fig. 12 Example of tracking failure on video sequence *EnterExitCrossingPaths1front*.

6 Conclusion

We proposed an effective and robust tracking approach based on soft-assigned BoF representation. In our approach, soft assignment is utilized to improve the robustness and discrimination of BoF representation in tracking. And tracking efficiency and effectiveness are also improved by patch sharing and adaptive patch size. The proposed approach is evaluated on the challenging video sequences with object occlusion, illumination variation and other complex conditions, and compared with seven state-of-the-art tracking methods. It shows that out approach obtains more accurate and stable tracking results.

In the future, we would like to combine global structure description in soft-assigned BoF representation to comprehensively describe the global and local information of the target. We will also pay attention to applying soft-assigned BoF tracking in video indexing and other applications.

Acknowledgements The authors want to thank the anonymous reviews for their helpful suggestion, and Tao Huang for his contribution in experiment. This paper is supported by Natural Science Foundation of China (61202320), Research Project of Excellent State Key Laboratory (61223003), and Natural Science Foundation of Jiangsu Province (BK2012304).

References

1. Zhong, W., Lu, H., and Yang M. H. Robust object tracking via sparsity-based collaborative model. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1838-1845. Providence, USA (2012)
2. Wang, M., Hong, R., Li, G., Zha, Z.J., Yan, S., and Chua T.S. Event driven web video summarization by tag localization and key-shot identification. IEEE Transactions on Multimedia. 14(4), 975-985 (2012)
3. Zhang, P., Thomas, T., and Emmanuel S. Privacy enabled video surveillance using a two state Indicatov tracking algorithm. Multimedia Systems. 18(2), 175-199 (2012)
4. Wachs, J.P., Kölsch, M., Stern, H., and Edan, Y. Vision-based hand-gesture applications. Communications of the ACM, 54(2), 60-71 (2011)
5. Wang, M., Hua, X. S., Hong, R., Tang, J., Qi, G. J., and Song, Y. Unified video annotation via multigraph learning. IEEE Transactions on Circuits and Systems for Video Technology. 19(5), 733-746 (2009)
6. Park, B. S., Yoo, S. J., Park, J. B., and Choi, Y. H. A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots. IEEE Transactions on Control Systems Technology. 18(5), 1199-1206 (2010)
7. Li, A., Tang, F., Guo, Y., and Tao H. Discriminative nonorthogonal binary subspace tracking. In: European Conference on Computer Vision, pp. 258-271. Heraklion, Greece (2010)
8. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., and Hengel, A. A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology (2013)

9. Ying, L., Xu, C., and Guo W. Extended MHT algorithm for multiple object tracking. In: International Conference on Internet Multimedia Computing and Service, pp. 75-79. Wuhan, China (2012)
10. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3360-3367. San Francisco, USA (2010)
11. Jiang, Y. G., Ngo, C. W., and Yang, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In: ACM International Conference on Image and Video Retrieval, pp. 494-501. Amsterdam, The Netherlands (2007)
12. Jégou, H., Douze, M., and Schmid, C. Improving bag-of-features for large scale image search. International Journal of Computer Vision. 87(3), 316-336 (2010)
13. Yang, F., Lu, H., and Chen, Y. Bag of features tracking. In: International Conference on Pattern Recognition, pp. 153-156. Istanbul, Turkey (2010)
14. Yang, F., Lu, H., Zhang, W., and Yang, G. Visual tracking via bag of features. IET Image Processing. 6(2), 115-128 (2012)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. Anchorage, USA (2008)
16. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In: European Conference on Computer Vision, pp. 661-675. Copenhagen, Denmark (2002)
17. Allili, M. S., and Ziou, D. Object of interest segmentation and tracking by using feature selection and active contours. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. Minneapolis, USA (2007)
18. Santner, J., Leistner, C., Saffari, A., Pock, T., and Bischof, H. Prost: Parallel robust online simple tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 723-730. San Francisco, USA (2010)
19. Hu, W., Li, X., Zhang, X., Shi, X., Maybank, S., and Zhang, Z. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. International Journal of Computer Vision. 91(3), 303-327 (2011)
20. Kim, Z. Real time object tracking based on dynamic feature grouping with background subtraction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. Anchorage, USA (2008)
21. Zhou, H., Yuan, Y., and Shi, C. Object tracking using SIFT features and mean shift. Computer Vision and Image Understanding. 113(3), 345-352 (2009)
22. He, W., Yamashita, T., Lu, H., and Lao, S. Surf tracking. In: IEEE International Conference on Computer Vision, pp. 1586-1592. Kyoto, Japan (2009)
23. Wang, S., Lu, H., Yang, F., and Yang, M. H. Superpixel tracking. In: IEEE International Conference on Computer Vision, pp. 1323-1330. Barcelona, Spain (2011)
24. Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision, pp. 1-22. Prague, Czech Republic (2004)
25. Nowak, E., Jurie, F., and Triggs, B. Sampling strategies for bag-of-features image classification. In: European Conference on Computer Vision, pp. 490-503. Graz, Austria (2006)
26. Lowe, D. G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 60(2), 91-110 (2004)
27. Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (SURF). Computer vision and image understanding. 110(3), 346-359 (2008)
28. Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., and Yamada, A. Color and texture descriptors. IEEE Transactions on Circuits and Systems for Video Technology. 11(6), 703-715 (2001)
29. Fei-Fei, L., and Perona, P. A bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Vision and Pattern Recognition, pp. 524-531. San Diego, USA (2005)
30. Wang, M., Hua, X. S., Tang, J., and Hong, R. Beyond distance measurement: Constructing neighborhood similarity for video annotation. IEEE Transactions on Multimedia. 11(3), 465-476 (2009)

31. Jiang, Y. G., Yang, J., Ngo, C. W., and Hauptmann, A. G. Representations of keypoint-based semantic concept detection: A comprehensive study. IEEE Transactions on Multimedia. 12(1), 42-53 (2010)
32. Gemert, J. C., Geusebroek, J. M., Veenman, C. J., Smeulders A. W. M. Kernel codebooks for scene categorization. In: European Conference on Computer Vision, pp. 696-709. Marseille, France (2008)
33. Zhu, S., Wang, G., Ngo, C. W., and Jiang, Y. G. On the sampling of web images for learning visual concept classifiers. In: ACM International Conference on Image and Video Retrieval, pp. 50-57. Xi'an, China (2010)
34. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(7), 881-892 (2002)
35. Ojala, T., Pietikainen, M., and Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(7), 971-987 (2002)
36. Adam, A., Rivlin, E., and Shimshoni, I. Robust fragments-based tracking using the integral histogram. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 798-805. New York, USA (2006)
37. Ross, D., Lim, J., Lin, R., and Yang, M. Incremental learning for robust visual tracking. In: International Journal of Computer Vision. 77(1-3), 125-141 (2008)
38. Babenko, B., Yang, M. H., and Belongie, S. Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 983-990. Miami, USA (2009).
39. Kwon, J., and Lee, K. M. Visual tracking decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1269-1276. San Francisco, USA (2010).
40. Kalal, Z., Mikolajczyk, K., and Matas, J. Tracking learning detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 34(7), 1409-1422 (2012)
41. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (VOC) challenge. International journal of computer vision. 88(2), 303-338 (2010)
42. Qiu, Z., Yu, T., Ren, T., Liu, Y., and Bei, J. Soft-assigned bag of features tracking. In: International Conference on Internet Multimedia Computing and Service. Huangshan, China (2013)
43. CAVIAR. http://homepages.inf.ed.ac.uk/rbf/caviar/.

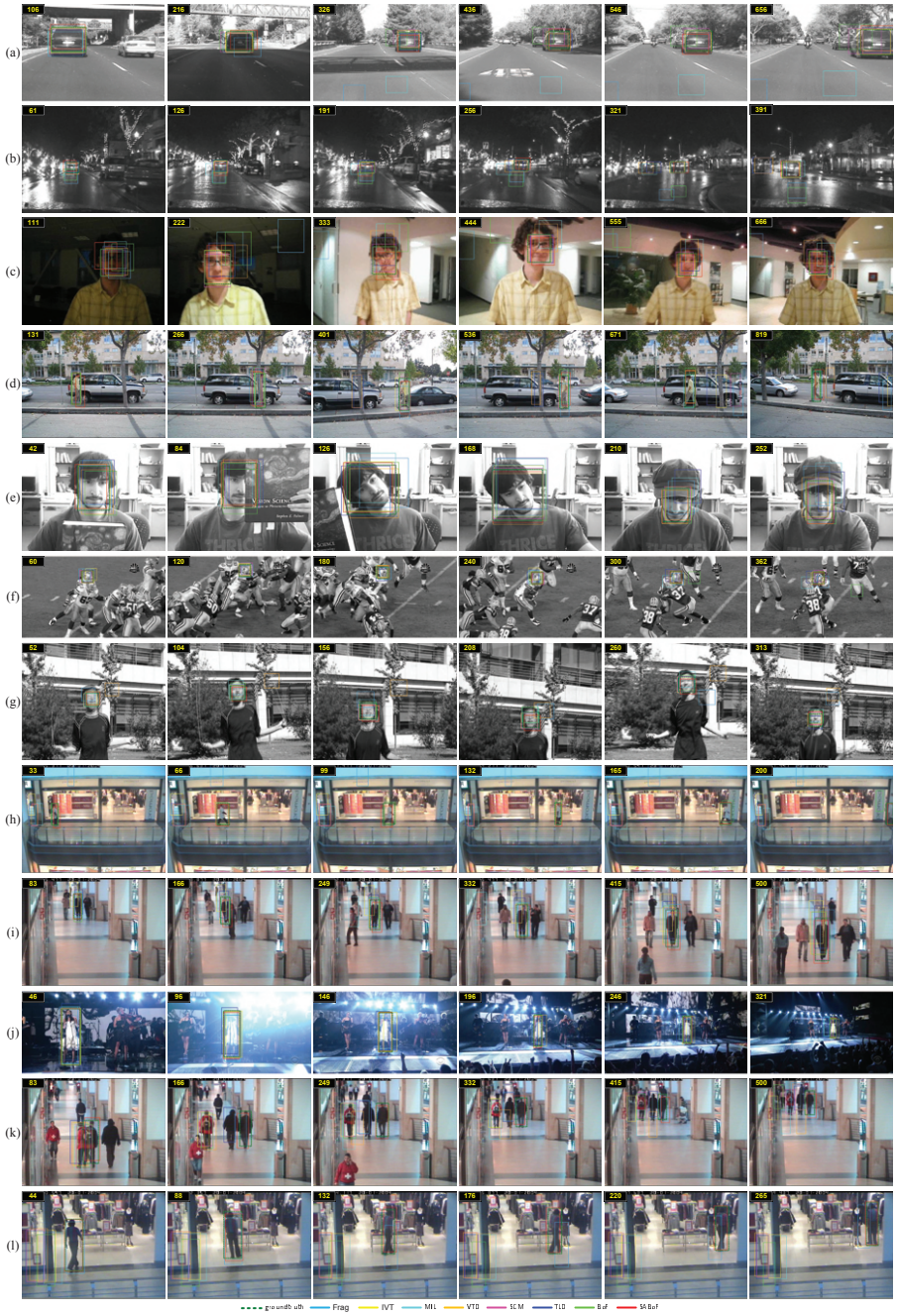

Fig. 13 Comparison of tracking results of our approach and other seven tracking methods. (a)-(l) Tracking results on video sequence *Car4*, *Car11*, *DavidIndoor*, *DavidOutdoor*, *FaceOccu2*, *Football*, *Jumping*, *OneStopEnter1front*, *ShopAssistant2cor*, *Singer1*, *ThreePastShop2cor* and *WalkByShop1front*, respectively.