Learning structured visual dictionary for object tracking

Fan Yang, Huchuan Lu, Ming-Hsuan Yang

Abstract

In this paper, we propose a visual tracking algorithm by incorporating the appearance information gathered from two collaborative feature sets and exploiting its geometric structures. A structured visual dictionary (SVD) can be learned from both appearance and geometric structure, thereby enhancing its discriminative strength between the foreground object and the background. Experimental results show that the proposed tracking algorithm using SVD (SVDTrack) performs favorably against the state-of-the-art methods.

1. Introduction

Numerous effective tracking approaches have been proposed in the literature [1]. One of the most important factors for effective tracking algorithms is how to account for appearance variation with effective representations. Among the existing algorithms, some focus on holistic appearance models [2–6], whereas others resort to local parts or patches [7–12].

In visual categorization, the bag-of-features (BoF) algorithm has been successfully applied to object and natural scene classification with different codebooks or decision criteria. While most existing works [13–17] focus on local parts or regions, SIFT descriptors [18] for object representation, Nowak et al. [19] show that random sampling of image patches with more discriminative power than that with interest point detectors when a large number of patches are used.

Aiming to develop an effective model with features extracted from image patches, we explore the BoF paradigm to collect representative features that best characterize an object and the background. We cluster a large set of patches into a compact number of groups as a visual dictionary which consists of the most distinctive features of both the foreground object and the background. Although Yang et al. [20] have applied BoF representation to tracking, their generative approach does not use negative samples or geometrical structures. The conventional BoF algorithm used in [20] treats an object image as a collection of independent patches without considering their spatial relationships. Numerous geometric descriptors have been developed to encode spatial information as features for object recognition. Cootes et al. [21] propose the active shape models (ASMs) to describe an object in terms of its contour. The ASMs consist of a set of landmark points representing the shape, which are analogous to the active contour model [22]. The active appearance models (AAMs) to describe an object [23]. Shape matching is carried out via an optimization process to minimize the difference between the current estimate of appearance and the target image. However, it is not applicable to real-time tracking since the optimization process is time-consuming. Recently, several algorithms have been proposed to incorporate spatial information with the patch collections [24–26] for object categorization and recognition. Nevertheless, these methods require significant numbers of training samples since they need to find statistically the most representative keywords to handle intra-class variance. Moreover, the time-consuming training procedures to describe the spatial relationships among the large training samples and build classification models make the previous works impractical for real-time object tracking. For object tracking, it is desirable to develop effective methods that exploit spatial relationships and constraints among features.

Within the context of object tracking, numerous methods that utilize spatial information have been proposed due to its importance. Some approaches are formulated based on Markov networks [27,28] to encode the relationships of interest points or regions in the image space. However, [27] uses a fixed number of local patches and in [28] both appearance and spatial features entirely rely on the locations of detected keypoints. Therefore, they are not flexible enough to deal with complex scenarios. Similarly, [29,30] also use keypoints to extract local appearance patches, but no local spatial information among keypoints is utilized although [29] takes the overall geometry of the entire object into consideration. In addition, many approaches rely on the motion estimation of interest points (based on features or regions) between successive frames.
These methods first match features between two frames and then estimate their corresponding motion parameters. The global object movement is then computed from the motion parameters of local features. However, features in these methods are processed independently and thus rich geometric structures among them within the same image are ignored. Here we propose a method that focuses on statistical relationships of features and does not involve motion estimation.

In this paper, we propose a tracking algorithm based on structured visual dictionary that exploits both appearance and geometric information. The contributions of this paper are summarized as follows. First, we extend the BoF representation scheme in the context of object tracking. We propose a discriminative tracking algorithm for learning an effective and compact visual dictionary of both foreground target and background. Second, we encode and exploit geometric constraints of local invariant features. Instead of computing movement of individual features to infer the global object motion, we model the statistical distribution of features in the 2D image plane. Third, our formulation facilitates the integration of appearance and geometric information. Both appearance and geometric information are characterized as histograms so they can be combined in a unified and reasonable way. In addition, the proposed formulation accommodates an update scheme easily, thereby facilitating effective object tracking.

2. Proposed algorithm

2.1. Initialization

In the first frame, the target object is initialized manually by a bounding box. We use the similarity transform to describe object motion with state variable $\chi_t$. A small number of training images are collected to construct the structured visual dictionary for the target object as follows. For each training frame, a rectangular bounding box enclosing the target object is bootstrapped either by another tracker (with visual inspection) or by hand. Similar to the sampling scheme in [34], we crop out a set of patches $P_\alpha = \{P\mid l(P) - l(\chi_t) < \alpha\}$ of random size within the object as positive samples, where $l(P)$ and $l(\chi_t)$ denote the locations of small patch $P$ and the object $\chi_t$ at time $t$. The distance $\alpha$ is a threshold to ensure all patches fall into the object region. For negative samples, we sample another set of patches $P_\beta = \{P\mid \beta < l(P) - l(\chi_t) < \gamma\}$ from the area outside the target. Here the distance $\beta$ and $\gamma$ are thresholds, and $\beta$ is larger than $\alpha$. These parameters are selected by imposing a range constraint on the central locations of patches such that these patches do not overlap with the object region and are not far away from the object. With several training frames, sufficient positive and negative samples are collected to build the structured visual dictionary at the outset of the tracking process.

2.2. Structured visual dictionary

In this section, we describe the statistical properties learned from the sampled patch sets. In contrast to conventional BoF approaches that use only appearance or texture to represent objects, we incorporate geometric information into the visual vocabulary, and refer to it as the structured visual dictionary (SVD).

2.2.1. Appearance keywords

Given two sets of patches $P^+$ and $P^-$, descriptors are extracted $f(P^+)$ and $f(P^-)$ where $f$ is a feature function. Next, these features are grouped into clusters $K_f$.

$$K_f = \{c(f(P^+)), c(f(P^-))\},$$

where $c$ denotes a clustering function that returns the center and $K_f = \{K_{f,m}\}_{m=1}^{M}$ is a set of cluster centers that are used as appearance keywords of the dictionary, where $M$ is the number of cluster centers.

Note that positive and negative samples are clustered separately in order to better contrast the learned appearance keywords between positive and negative sets. We note that in [20] only positive samples are used to learn visual dictionary, which explains why the tracker is sensitive to visual drift.

To better represent objects, the proposed SVD consists of two sets of appearance keywords obtained from the RGB descriptor and the local binary pattern (LBP) descriptor [35] which are extracted from randomly sampled patches instead of using interest point operators. Therefore, we describe an object from two different aspects which uncover as many representative features as possible. These collaborative sets of keywords derived from two descriptors often complement each other for more effective object representations, thereby facilitating the tracking process.

2.2.2. Geometric correlations

Aside from appearance keywords, we incorporate geometric constraints of features in the proposed SVD. While some existing methods have utilized geometric information of features [24–26] in the context of object categorization and recognition, these methods are not efficient or effective for object tracking. Within the context of visual tracking, several methods [31,36,28,32,33] that use interest points for describing target objects have been proposed.

Tracking is then carried out by matching individual interest points in consecutive frames. In other words, such approaches treat all the features independently and do not take the spatial relationships of neighboring features into account. However, the geometric information among neighboring features preserves important cues for feature matching and object tracking. Therefore, we utilize the relative positions of detected interest points in a 2D plane to incorporate geometric information for constructing a structured visual dictionary. Our motivation is that the spatial distribution of interest points within an object conforms with a stable pattern which can be statistically described by modeling the spatial relationships among these points. Note that the way of encoding spatial information does not depend on the correspondence of interest points in successive frames.

We detect interest points from an image and extract their SIFT features [18]. As shown in Fig. 2, the relative positions of stable interest points extracted from an object in successive frames almost remain constant. The histograms of the five images from the target object in Fig. 2 are relatively similar. That is, the geometric structure of stable interest points are not affected much by smooth motion, scale change and rotation in successive frames. In contrast, interest points of the regions from the background cannot be modeled well by a specific distribution (See Fig. 2).

Motivated by these observations, we propose an efficient method to exploit geometric structure as discriminative features for object tracking. Inspired by the shape context approach [37], we consider local statistics originating from the center $c$ of the extracted object region to all the other interest points.

Assume that we detect $R$ interest points from a target object, we use $V_r$ to represent a vector from $c$ to the $r$-th point. The set of local statistics $V = \{V_r\}_{r=1}^R$ describes the positions and distances of interest points with respect to $c$. To obtain a compact and robust representation, we quantize local statistics in $V$ in the polar coordinate system as shown in Fig. 1. Suppose that we divide the plane of polar coordinates into $U_m \times U_a$ regions where $U_m$ and $U_a$ represent different magnitudes and angles. For a specific region $(u_m, u_a)$, where $u_m$ and $u_a$ are the indices of magnitude bin and angle bin respectively, we count the number of interest points within this grid, thereby constructing a matrix $M(u_m, u_a)$ which records the distribution of $V$ (See Fig. 1). We then convert each matrix $M(u_m, u_a)$ to a histogram $H(u)$, which preserves compact spatial information of interest points. Each bin $u$ of the histogram represents a specific magnitude and angle region. In practice, we use 4 bins and 5 bins for quantization of magnitude and angle, respectively. The motivation to use histogram representation for spatial information is to facilitate
update and fusion with appearance information in a simple and consistent way.

We also use the same method to compute the spatial histogram of interest points from the background, as shown in Fig. 2. However, since interest points of sampled regions from the background may appear in any configurations, their spatial histograms do not contain representative information. The appearance keywords and the geometric correlations of interest points together are the integral components of the proposed SVD for object tracking.

### 2.3. Image representation with histograms

Once we have the SVD, an image region can be represented by two appearance histograms and a spatial histogram. Within each image region, a set of patches of random sizes \( P \) are randomly sampled. The appearance histogram is then created by assigning each patch in \( P \) to its nearest keyword in \( K \) and computing the occurrence frequency of corresponding keyword within this image region. The spatial histogram characterizes the spatial distribution of a set of interest points. The formulation of spatial histogram is mathematically consistent with appearance histograms, because both are histograms describing the statistical information (appearance or location) of a group of representative features.

For a set of keywords \( K \), we arrange positive and negative keywords separately, and then concatenate bins of indices from two kinds of keywords. Subsequently, the object is represented by an ordered appearance histogram \( H_f(u) \), where \( u \) indicates a bin of \( H_f(u) \). Assume that \( H_f(u) \) has \( U \) bins, so the first \( U/2 \) bins record the occurrence of positive keywords and the remaining \( U/2 \) bins represent that of negative ones. Moreover, we use the following equation rather than raw counts to compute \( H_f(u) \):

\[
H_f(u) = \sum_{n=1}^{N} \exp(-d_u/\sigma) \cdot \delta(b(P_n)-u)
\]

where \( H_f \) denotes the appearance histogram regarding feature \( f \) and \( P_n \) denotes a patch. In the equation above, \( N \) is the number of patches within a candidate, \( b(\cdot) \) indicates the index of bin, and \( \delta(\cdot) \) is a delta response function that equals 1 when \( b(P_n) \) equals \( u \). By multiplying a weighting term \( \exp(-d_u/\sigma) \), the patch more similar to the keyword contributes more to the corresponding bin. In the weighting term, \( d_u \) is the minimal Euclidean distance between the feature of a patch \( f(P_n) \) and the closest keyword in \( K_f \) (\( \sigma \) is set to 0.2 in our experiments).

#### 2.4. Object tracking with SVD

At time \( t \), we draw \( L \) candidate states \( X_t = \{x_t\}^{L}_{l=1} \) where \( x_t = (c, d, c, y, s, \theta) \) around the tracked object at time \( t-1 \) using random walks with factorized Gaussians in a way similar to [38]. \( c, d, c, y, s, \theta \) represent the center, scale and rotation angle of the state \( x_t \). Each state \( x_t \) corresponds to a rectangular image region enclosed by a bounding box. Within each bounding box, we randomly sample small patches \( P_c = \{P_{n}\}^{N}_{n=1} \) of random size and extract their RGB features \( f_1(P) \) and LBP features \( f_2(P) \). After computing the distances between features of patches \( f(P_c) \) and keywords \( K_f \), the image patch region at state \( x_t^{st} \) is represented by two appearance histograms \( H_{f1} \) and \( H_{f2} \). We also detect interest points in the image patch at state \( x_t^{st} \) and compute the spatial histogram \( H_{f} \) of these points. Next, we compute the histogram similarity using the appearance and spatial histograms by:

\[
C_{w} = \max_{w} \exp \left(-d_{x}(H_{w}, H_{w})\right), \quad \forall w = f_1, f_2, s.
\]

where \( H_{w} \) is an appearance histogram or spatial histogram of the candidate sample. \( H_{w} \) is a set of corresponding histograms computed from the training images, and \( d_{x} \) is a function that computes \( \chi^2 \)-distance. Note that no similarity regarding individual histograms is computed, and we focus more on the statistical properties of the group of randomly sampled patches. As summarized in Algorithm 1, we obtain similarities of each sampled candidate region with respect to the target object using appearance and spatial histograms.

![Fig. 1. Formulation of geometric information of SVD. Left: the pattern of detected points where c is the center of a normalized image. Locations of points are represented in polar coordinates with quantized magnitude and angle. Each bin is a quantized region counting the number of points inside this region. The numbers of bins for magnitude and angle are 4 and 5. Right: matrix characterizing the distribution of interest points.](image1)

![Fig. 2. Examples of detected interest points in the target object and background regions. We use 5 samples for the target object and 5 samples for the background. Sets of spatial histograms of positive samples and negative samples are shown in the last image in each row, enclosed by red bounding box and blue bounding box respectively. Each row of the histograms represents a histogram for each sample image. Notice the relatively stable pattern of positive histograms.](image2)
Algorithm 1. SVD matching (w.r.t. one candidate region at $x_t$)

Given: $K_R$ and $H_2$ for RGB feature, $K_L$ and $H_2$ for LBP feature, spatial histograms $H_f$, SIFT features $f_i$ of detected interest points from a positive training sample (randomly selected) or a previous tracked object.

Input: Set of patches $P = \{P_1, \ldots, P_N\}$

1. for $n = 1$ to $N$
   
   (a) Find $d_{f_1, n} = \min(f_1(P_n), K_{f_1})$ for RGB and index of the keyword $u_{f_1, n}$
   
   (b) Find $d_{f_2, n} = \min(f_2(P_n), K_{f_2})$ for LBP and index of the keyword $u_{f_2, n}$

2. end for

3. Construct appearance histograms $H_f$ and $H_2$ using (2)

4. Detect interest points in candidate region at state $x_t$

5. Extract SIFT features to perform interest point matching with $f_i$

6. Construct spatial histogram of matched points $H_t$.

7. Determine appearance similarities $C_{f_1}$ and $C_{f_2}$ of two features and spatial similarity $C_s$ using (3)

Output: Similarities $C_{f_1}$, $C_{f_2}$ and $C_s$ between a candidate region and trained samples.

We note that it is less effective to simply compute the spatial histogram from all detected points. If detected points are from the background but happen to have similar geometric relationships to those of the foreground, the computed similarity value $C_s$ of the candidate region may be high and leads to inaccurate results. Therefore, we add a constraint for feature matching. In a candidate region, only points that match those from the previously tracked object are used for constructing $H_t$, whereas other points lying in the background region are discarded. Despite its effectiveness, this strategy may discard some new interest points due to the appearance change of the target object. This issue is addressed by an update scheme discussed in Section 2.5.

The appearance similarities $C_{f_1}$ and $C_{f_2}$, and spatial similarity $C_s$ for a candidate region are integrated by:

$$C_j = \exp \left( \frac{C_{f_1}}{\sigma_{f_1}}^2 + \frac{C_{f_2}}{\sigma_{f_2}}^2 + \frac{C_s}{\sigma_s} \right).$$

where $\sigma_{f_1}$, $\sigma_{f_2}$, and $\sigma_s$ are scaling terms that are set empirically to 0.02 and fixed in our experiments. The reliability of different features is computed by histogram similarities with the combination of appearance and spatial information. Thus, the proposed tracker is less likely to drift from the target as long as at least one feature is extracted to provide sufficient information of the target object, as shown in our experiments. A number of candidate regions with highest joint similarities are weighted and considered as the state of the target object at time $t$ by

$$S_t = \sum_{i=1}^{L'} C_{i,j} x'_i / \sum_{i=1}^{L'} C_{i,j}$$

where $L'$ is used to determine how many candidates are used (e.g., $L'$ equals 5% of $L$ candidate states in our experiments), and $C_{i,j}$ is the joint similarity for candidate $x'_i$ using (4).

2.5. Online update

Rather than a constant and off-line learned appearance model, all the histograms in the SVD are updated to account for appearance change of the target object.

2.5.1. Appearance histogram update

At each time $t$, we store the state $S_t$ of the tracked object and the corresponding accumulative similarity value $Z_t = \sum_{i=1}^{L'} C_{i,j}$. Large $Z_t$ means that all selected candidates have high probabilities of being the target. Thus, during the period up to the next update at time $t + T$, we have a set of states for the tracked objects $\{S_t, Z_t\}_{t+T}^T$, from which we choose the $S_t$ with highest $Z_t$. From the image patch at $S_t$, we construct two appearance histograms based on the RGB and LBP features, and update the corresponding set of trained appearance histograms $H_f$ and $H_2$, by discarding the oldest histograms respectively. We choose the histogram that best characterizes the target object to account for the appearance change. The update period $T$ is predetermined to accommodate typical object motion so that this scheme is not quick to adapt to noise or too slow to respond to the change of the target objects. We show the change of appearance histograms that account for the foreground and background change of a test sequence in Fig. 3.

2.5.2. Spatial histogram update

The update process of a spatial histogram is slightly different from that of an appearance histogram. At each update, the matching process determines a number of interest points from a candidate which are used for computing a spatial histogram. Assume that $p$ detected interest points are stored from the previously tracked object and $q$ points from a candidate $S_t$ at time $t$, then $r$ matched points ($r \leq \min(p, q)$) are determined. If $r > 0$, where $\omega$ is a percentage (e.g., it is set to 80% in our experiments), it means that the tracking result does not deviate much from the target location. Thus, it is reliable to utilize the current tracking result to update the spatial histogram. Similar to the scheme used for the appearance histogram update, new histograms are added to the set and old ones are discarded. When spatial histograms are updated, the corresponding features of old interest points are replaced with those of the new ones for subsequent tracking process. The update process is illustrated in Fig. 4.

In Fig. 5, we show some results of matched interest points for the spatial histogram update and tracked objects of the tom2 image sequence. The change of three similarity values is also presented in Fig. 7. This figure clearly demonstrates that the drifts result in few matched points, i.e., low spatial similarity values (see frame #289). Even when there are no matched interest points in some frames, our method is still able to compute the spatial histogram (all bins are 0) and similarity values used for the overall similarity in (4).

Note that the appearance and spatial histograms are updated separately because they do not change simultaneously in most cases. If the appearance of the object changes but there are only few interest points, the spatial histograms do not contribute to the combined similarity much. Therefore, we update only the appearance histograms, but there is no need to update the spatial histograms. Experimental results validate the effectiveness of the update mechanism.

3. Experiments and results

We present experimental results with comparisons to four state-of-the-art tracking algorithms and discussions in this section.

3.1. Experimental setups

We evaluate the proposed tracking algorithm with SVD (referred to as SVDTrack) using several challenging image sequences. The SVDTrack system is implemented by Matlab without code optimization and takes 10s to process a frame on a PC with 2.1 GHz CPU and 4GB memory. In fact, the keypoint detection takes the most time while the histogram construction and comparison are very efficient. The values of parameters are set empirically and listed as follows. The number of sampled candidates $N$ is set to 200 in all the experiments. Each candidate region is normalized block of 64 $\times$ 64 pixels, from which 50 patches ranging from 8 $\times$ 8 to 16 $\times$ 16 pixels are randomly selected. For each patch, we extract a 64-dimensional RGB feature vector by quantizing each channel into 4 bins, and a 59-dimensional LBP feature vector using uniform LBP descriptor with 8 neighbors [35]. Both the numbers of positive and negative keywords are set to 10, and the appearance histogram for each feature has 20 bins. As appearance variance of the target object is much less than that of the background in consecutive frames, a small number of keywords is sufficient for object tracking. We detect interest
points in each 64 × 64 candidate region, compute their relative positions with its center, and construct the spatial histogram that is normalized using $l_1$-norm. At the outset of the tracking process, the first 5 frames of an image sequence are used for training appearance keywords and spatial histograms. The update period $T$ for appearance histogram update is set to 5, 10 or 20 empirically to achieve a trade-off between representativeness and adaptive ability of the tracker. On the other hand, the update period for spatial histograms is entirely determined by the number of interest points during tracking.

We evaluate the proposed SVDTrack method in terms of its capability in handling occlusion, pose and appearance change for object tracking. We compare it with four state-of-the-art tracking methods, the FragTrack [8], TLD [39], MILTrack [34] and VTD tracking [6] algorithms, with the provided source codes for fair performance evaluation. The FragTrack method is designed to handle occlusion by using a template of multiple image fragments for object representation. Our approach is similar in spirit to this method as both utilize local information from a group of small patches. The MILTrack has been shown to be able to handle visual drifts. The TLD tracking method, which simultaneously tracks the object, learns its appearance and detects it whenever it appears in the video, is able to handle large illumination change. Finally, the VTD method, using sparse PCA algorithm to select the best template in each frame, is able to deal with large motion and lighting changes. In our experiments, the search region of FragTrack ranges from 10 × 10 to 30 × 30 pixels with 16-bin grayscale histogram. The patch size in the TLD method remains 20 × 20 pixels, and the VTD method uses 600 candidate templates. Note that except some parameters specific to individual trackers, all the common parameters are fixed for all tracking methods on each test sequence. All the source codes and data sets will be made available to the public.

3.2. Empirical results

We present and discuss tracking results of our approach and other state-of-the-art methods. The test image sequences include all major challenging issues in object tracking: heavy occlusion, illumination change, fast motion, low contrast, camera movement, and pose variation. More tracking results are available at http://www.youtube.com/svdttrack.

3.2.1. Occlusion

Fig. 6(a), (b), and (c) show the tracking results on the woman sequence from [8], and two videos, face1 and dance3, from our own dataset. Since the clothes of the person have very similar appearance to the top of each frame, is able to deal with large motion and lighting changes. In our experiments, the search region of FragTrack ranges from 10 × 10 to 30 × 30 pixels with 16-bin grayscale histogram. The patch size in the TLD method remains 20 × 20 pixels, and the VTD method uses 600 candidate templates. Note that except some parameters specific to individual trackers, all the common parameters are fixed for all tracking methods on each test sequence. All the source codes and data sets will be made available to the public.

3.2. Empirical results

We present and discuss tracking results of our approach and other state-of-the-art methods. The test image sequences include all major challenging issues in object tracking: heavy occlusion, illumination change, fast motion, low contrast, camera movement, and pose variation. More tracking results are available at http://www.youtube.com/svdttrack.

3.2.1. Occlusion

Fig. 6(a), (b), and (c) show the tracking results on the woman sequence from [8], and two videos, face1 and dance3, from our own dataset. Since the clothes of the person have very similar appearance to the top of
the white sedan, this sequence poses significant challenges for object tracking when the subject is partially occluded. Thus, methods relying on global templates (MILTrack and VTD) are not able to handle such scenarios. It is worth noticing that in [8] a smaller tracking window is used (that encloses about two-thirds of the full body). However, if we enlarge the window to enclose the entire human body, the FragTrack method easily loses track of the subject because many background pixels are inevitably considered as parts of the foreground object. In contrast, our SVDTrack method keeps track of the woman well and outperforms all other methods. Although some parts of the target object are invisible in some frames, there are still sufficient representative features that can be used for patch-based tracking methods. In addition, since some interest points can still be detected, the relative positions to the central point of the target object can be exploited. That is, geometric information encoded in our spatial histograms facilitates our tracker to capture the stable geometric distribution of interest points in such scenarios, and thus renders robust results.

The tracking results using the dance3 sequence from our own dataset are presented in Fig. 6(c) where two dancers are occluded by each other. The dancers also change their poses frequently. All other methods fail when the female dancer is fully occluded by the male dancer for a short time, but only our tracker continues to track her when she reappears. Likewise, the results on the face1 sequence (Fig. 6(b)) show that our method is able to track the girl as she swings her head while being heavily occluded.

3.2.2. Pose variation

Fig. 6(d) shows the tracking results of the bird2 sequence in which a bird walks back and forth while undergoing large pose variation. Note that the target is also partially occluded at times with shape deformation. The background pixels are likely being considered as parts of foreground objects for template-based methods that rely on rectangles to enclose target objects (unless some mechanisms are introduced to separate foreground and background pixels). Our SVDTrack algorithm successfully keeps track of the bird in this sequence, while other methods render inaccurate results, e.g., the VTD method drifts away and the TLD tracker fails when the bird is partially occluded. The MILTrack method is equipped with a detection mechanism, it fails to adapt the model for tracking the target object in this sequence.

3.2.3. Fast motion

In Fig. 6(g), we present the tracking results of the bolt sequence. Due to fast acceleration and motion of the target athlete, it is difficult for trackers to precisely locate his position. In addition, the camera view change makes this sequence rather challenging. Our SVDTrack algorithm is able to track the athlete throughout the entire sequence, while other methods all drift at the beginning or when the camera angle changes. For instance, the VTD, TLD and MILTrack methods drift away when the athlete just begins to accelerate. As a result of using the proposed SVD, our tracker is able to track fast moving objects in a complex scene with a camera view change.

The tracking results with the tom2 and tom3 sequences are shown in Fig. 6(h) and (i), in which the target objects undergo fast and drastic movements with partial occlusion. The proposed SVDTrack algorithm performs well against other methods, as they either drift (MILTrack, FragTrack and VTD) or fail (TLD) at times.

3.2.4. Illumination and scale change

In the cup sequence, the target object undergoes illumination and scale change, as shown in Fig. 6(j). The FragTrack and MILTrack methods do not handle illumination change or larger scale variation well, whereas our SVDTrack method is able to deal with such situations due to the use of LBP features. The MILTrack method gradually drifts away from the target when there is a large scale change. Although the TLD method is equipped with a detection mechanism, it fails to adapt the model for tracking the target object in this sequence.

The average tracking errors in terms of the central object position is presented in Table 1 with manually labeled ground truth data.

3.3. Discussions

To evaluate the importance of individual terms in [4], we remove one component and run the SVD tracker on all test sequences, resulting in three trackers which are SVDTrack without LBP feature, SVDTrack without RGB feature and SVDTrack without geometric structure (referred as SVD1, SVD2 and SVD3 respectively). The qualitative and quantitative comparisons are shown in Fig. 6 and Table 1. On most sequences, the geometric information helps to improve the tracking performance, especially when the target object undergoes occlusion, large pose change and shape deformation. In some cases, the appearance information is enough for good tracking results, so the importance of geometric information
decreases especially when the geometric information is not distinct (the cup sequence). The importance of the individual terms is also reflected in Fig. 7.

Overall, our approach performs favorably in all sequences with lower errors and deviations. In our method, appearance keywords preserve the most representative features of the foreground and background, and spatial histograms are stable to facilitate object tracking. Thus, our tracker does not easily drift away from the target object. Additionally, we evaluate the effectiveness of online update by showing the change of similarity values of RGB feature, LBP feature and spatial information for sequences tom2 and cup in Fig. 7. It is obvious that the similarity values change throughout the sequence during tracking. Specifically, our tracker successfully captures the change of the object and adjusts the importance of individual features accordingly.

Table 1
Average tracking error (in pixels). The error for each frame is computed by measuring the difference of the center location of the object between the ground truth and the tracking result. Italic font means the best and bold font means the second best.

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Frag</th>
<th>MIL</th>
<th>TLD</th>
<th>VTD</th>
<th>SVD</th>
<th>SVD1</th>
<th>SVD2</th>
<th>SVD3</th>
</tr>
</thead>
<tbody>
<tr>
<td>bird2</td>
<td>30</td>
<td>19</td>
<td>87</td>
<td>55</td>
<td>21</td>
<td>125</td>
<td>50</td>
<td>28</td>
</tr>
<tr>
<td>face1</td>
<td>8</td>
<td>77</td>
<td>3</td>
<td>16</td>
<td>6</td>
<td>17</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td>woman</td>
<td>107</td>
<td>109</td>
<td>95</td>
<td>104</td>
<td>10</td>
<td>127</td>
<td>99</td>
<td>127</td>
</tr>
<tr>
<td>bolt</td>
<td>122</td>
<td>391</td>
<td>90</td>
<td>323</td>
<td>21</td>
<td>101</td>
<td>139</td>
<td>16</td>
</tr>
<tr>
<td>cup</td>
<td>28</td>
<td>20</td>
<td>48</td>
<td>8</td>
<td>12</td>
<td>21</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>tom2</td>
<td>30</td>
<td>38</td>
<td>63</td>
<td>206</td>
<td>21</td>
<td>79</td>
<td>70</td>
<td>104</td>
</tr>
<tr>
<td>tom3</td>
<td>38</td>
<td>59</td>
<td>22</td>
<td>6</td>
<td>13</td>
<td>19</td>
<td>65</td>
<td>13</td>
</tr>
<tr>
<td>dance3</td>
<td>32</td>
<td>57</td>
<td>53</td>
<td>29</td>
<td>19</td>
<td>106</td>
<td>127</td>
<td>87</td>
</tr>
<tr>
<td>david</td>
<td>187</td>
<td>39</td>
<td>15</td>
<td>69</td>
<td>15</td>
<td>22</td>
<td>25</td>
<td>33</td>
</tr>
<tr>
<td>sylvester</td>
<td>99</td>
<td>31</td>
<td>17</td>
<td>49</td>
<td>14</td>
<td>100</td>
<td>116</td>
<td>20</td>
</tr>
</tbody>
</table>

Fig. 6. Tracking results of on all test sequences. We also evaluate three additional trackers which are SVDTrack without LBP feature, SVDTrack without RGB feature and SVDTrack without geometric structure (referred to as SVD1, SVD2 and SVD3 respectively). Please refer to Section 3.3 for details.
LBP feature are equally important. But when the lighting condition changes, the LBP feature becomes more important (as the colors of the cup and background fall into the same bin of the color histograms).

4. Conclusion

In this paper, we propose a tracking algorithm with structured visual dictionary (SVD) named SVDTrack. The proposed SVD exploits both appearance and geometric information. We exploit statistics of relative geometric information of local invariant features. Both represented as histograms, appearance and geometric features are integrated for object tracking, and they contribute to locate objects according to scene change. Experimental results show that our SVDTrack algorithm performs favorably in comparison to several state-of-the-art tracking methods. Our future work will focus on a discriminative classifier to classify positive and negative histograms rather than computing their \( \chi^2 \) distances. In addition, we will explore better histogram update strategies.

References