Visual tracking via context-aware local sparse appearance model

Guiji Li a, Manman Peng a, Ke Nai a, Zhiyong Li a, Keqin Li a, Keqin Li a,b

a College of Information Science and Engineering, Hunan University, Changsha 410082, Hunan, China
b Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Abstract

Most existing local sparse trackers are prone to drifting away as they do not make use of discriminative information of local patches. In this paper, we propose an effective context-aware local sparse appearance model to alleviate the drift problem caused by background clutter and occlusions. First, considering that different local patches should have different impacts on the likelihood computation, we present a novel Impact Allocation Strategy (IAS) with integration of the spatial-temporal context. Varying positive impact factors are adaptively assigned to different local patches based on their ability distinguishing the spatial context, which provides discriminative information to prevent the tracker from drifting. Furthermore, we exploit temporal context to introduce some historical information for more accurate locating. Second, we present a new patch-based dictionary update method being able to update each patch independently with the validation of effectiveness. On the one hand, we introduce sparsity concentration index to check whether the local patch to be updated is a valid local patch from the target object. On the other hand, spatial context is further employed to eliminate the effect of the background. Experimental results show the superiority and competitiveness of the proposed method on the benchmark data set compared to other state-of-the-art algorithms.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

As one of the most fundamental and important topics in pattern recognition and computer vision, visual tracking has received widespread concerns [1–9]. After obtaining the unique initial states of an uncertain object (maybe a person or a car), visual tracking aims at estimating the size and location of this specific object continuously in an image sequence. Although there are many breakthroughs in recent years, visual tracking still remains a quite challenging task due to the factors such as occlusion, varying illumination, background clutter, etc.

Recently, a large number of sparse representation based trackers have sprung up in visual tracking with demonstrated success [10]. The basic idea of sparse representation in tracking is to represent each target candidate by the linear combination of dictionary atoms with sparse constraint. According to the conventional representation scheme, sparse representation based appearance model can be classified into global or local pattern.

Global sparse trackers [11–16] adopt the holistic template of a target as the appearance representation. Although good performance are reported, their methods may fail with high possibility when the local appearance changes occur. Partial variations will cause an imprecise similarity measurement between candidates and the object since they are treated as single entities. Compared to global sparse appearance models, local sparse appearance models [17–22] are more attractive due to their effectiveness in handling partial occlusions. However, there still exist several drawbacks in local sparse appearance models as follows. (1) Different impacts of different local patches are not significantly considered on the likelihood computation. Due to the appearance variation difference of local patches during tracking, it is necessary to make a distinction among them rather than treating them equivalently. (2) Local sparse appearance models are less effective in dealing with background clutter. The underlying reason for this weakness lies in that discriminative information is rarely used in local sparse appearance models. (3) Most local sparse trackers update the dictionary based on the holistic pattern. That is to say, once the tracking result is updated, all local patches within it are updated naturally. When the tracking result contains some occluded patches, error will be accumulated if updated and thus result in a dirty dictionary. Otherwise, some effective appearance changes may fail to be captured, leading to degraded representation ability of the dictionary.

References


This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.

E-mail addresses: guiji.li@hnu.edu.cn (G. Li), pengmanman@hnu.edu.cn (M. Peng), naike_hnu@hnu.edu.cn (K. Nai), zhiyong.li@hnu.edu.cn (Z. Li), like@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.jvcir.2018.09.004

1047-3203/© 2018 Elsevier Inc. All rights reserved.
To address above issues, we propose a context-aware local sparse appearance model for robust visual tracking. First, considering that different local patches should have varying degrees of impact on the likelihood computation of target candidates, we develop a novel Impact Allocation Strategy to adaptively allocate positive impact factor for each local patch using spatial-temporal context. Specifically, a local patch which is more distinguishing from the spatial context will obtain higher positive impact factor on the final likelihood function. To this end, we separately construct local object dictionary and local context dictionary to represent each patch inside the tracking result, which introduces some discriminative information to alleviate the drift problem. Furthermore, we exploit temporal context for more robust tracking. Historical information provided by the previous frame is utilized to help locate the target more accurately. To the best of our knowledge, there are few tracking methods which integrate the spatial-temporal context information into local sparse appearance model to consider appearance difference among different local patches. And extensive experiments have demonstrated the effectiveness of this integration.

Second, to ensure that all effective appearance changes can be captured from the tracking result, a new patch-based dictionary update method is proposed in this paper. All local patches are updated independently with the validation of effectiveness using update method is proposed in this paper. All local patches are updated independently with the validation of effectiveness using sparsity concentration index [23] and spatial context. On the one hand, we exploit sparsity concentration index to check whether the local patch to be updated is a valid local patch from the target object. On the other hand, spatial context is used to provide some discriminative information to eliminate the effect of the background. In this way, even though most of local patches inside the target object undergo heavy occlusions, the tracker is still capable of capturing effective appearance changes from un-occluded local patches without missing.

The main contributions are summarized as follows.

1. An effective context-aware local sparse appearance model is proposed for robust tracking. The reliability of each local patch is measured through the whole tracking process with spatial-temporal context.
2. A novel impact allocation strategy (IAS) is developed to consider appearance difference among local patches. Different local patches are adaptively allocated varying positive impact factors on the likelihood computation of target candidates.
3. A patch-based dictionary update scheme is presented to ensure that all effective appearance changes can be updated into the dictionary without missing, even if the tracking result is mostly under occlusion.

2. Related works

There is extensive literature about various tracking methods, we advise readers to refer to [24–26] for thorough acquaintance. Here, we only talk about the most related works to ours and the tracking technology used in this study.

2.1. Part-based trackers

The part-based tracking methods become more popular depending on their robustness against partial appearance variations. Zhang et al. [27] propose a part matching tracker (PMT) for robust visual tracking. PMT matches parts from multiple adjacent frames via a locality-constrained low-rank sparse learning method. Lin [28], Li et al. develop a structured patch-based tracking method by simultaneously modeling the appearance of individual patches and the spatial information among them. Li et al. [29] attempt to identify and exploit the reliable patches to perform robust tracking. The final target location and scale are determined via a Hough voting-like scheme based on all reliable patches. Liu et al. [30] develop a real-time part-based tracker that learns multiple part correlation filters and combines their response maps with consideration of the reliability and temporal smoothness property of each part.

2.2. Sparse representation for tracking

Sparse representation is a popular technology that is widely used in computer vision task including face recognition [23,31] and visual tracking. In recent years, a variety of sparse representation-based trackers have been developed with significant success [2,11,12,14–17,20–22,32–34].

Global sparse trackers: Mei and Ling [15] represent each target candidate as a linear combination of target templates and trivial templates with sparse constraint in a particle filter framework, and then determine the tracking result by finding the candidate with the minimal reconstruction error using target templates. This is the first time sparse representation is applied to visual tracking. Zhang et al. [14] exploit the similarities among target candidates to enforce joint sparsity on their representations for handling tracking drift. Hong et al. [35] propose a multi-task multi-view joint sparse tracker with integrating richer feature representations to deal with various tracking challenges. Wang et al. [12] present a novel appearance model which exploits both incremental subspace learning and sparse representation scheme for robust object tracking. In order to improve the discriminative power of sparse coding coefficients at a low computation cost, Hong-tu et al. [16] decompose the origin sparse coding problem into two sub sparse coding problems and compute the final sparse representation by Cartesian product.

Local sparse trackers: Zhang et al. [21] make full use of the intrinsic relationship among and inside target candidates and the spatial layout structure of the local patches for robust tracking with efficient computation. In [17], a structural local sparse appearance model is developed by performing alignment-pooling operation on the sparse codes of all local patches. Zhao et al. [36] propose a dual-scale structural local sparse appearance model, where larger-scale model is used to capture the structural holistic feature of the target and the smaller-scale model is used to capture the structural local features of the target. Qi et al. [19] propose a structure-aware local sparse appearance model to obtain more powerful discriminative ability. In their work, both global and local sparsity constraints are applied to the sparse representation of target candidates. Zhong et al. [2] propose a sparse collaborative framework, where sparse generative model pays much attention to occluded local patches in terms of similarity measurement and template update. Ma et al. [20] design a robust tracking framework using a strong classifier and structural local sparse descriptors to deal with partial occlusion and deformation. To further take the difference among local patches into account, He et al. [22] turn to treat patches differently and emphasize the contributions of key patches which are determined by their location and the states of occlusion. Tian et al. [34] utilize the color similarities between the local patches from templates and candidates as weights of them for obtaining a more reliable tracker. Nai et al. [37] propose an effective local sparse appearance model to deeply explore the appearance characteristics of different local patches. They respectively define stable patches, valid patches and invalid patches and consider their different importance on the target locating.

Low-rank trackers: In [38], Zhang et al. propose a novel low-rank sparse tracker, which formulates the tracking problem as a
sparse and low-rank representation problem. Such joint representation (sparse and low-rank) in their formulation can be obtained with an attractive computational cost due to the inherent low-rank structure of particles. Sui et al. [39] make use of the generative and discriminative information to achieve impressive tracking performance via a discriminative low-rank learning method. Furthermore, they introduce a sparse additive residual error term to deal with occlusions. Yang et al. [40] propose a temporal restricted reverse-low-rank learning algorithm, which introduces the low-rank assumption into the successive target observations by representing the positive and background templates via the candidates. Different from above low-rank trackers, Zhou et al. [41] focus on learning a discriminative and low-rank dictionary without imposing any low-rank and sparse constraints on the coefficient matrix for robust tracking.

2.3. Context information for tracking

As an important clue in visual tracking, context provides additional information to help locate the object more accurately. Yang et al. [42] propose a context-aware collaborative tracking framework, in which the context (a set of auxiliary objects with some common properties) together with the target is tracked for further verification. In [43], local context is defined and exploited to help locate the target. When the appearance of the target object undergoes significant changes, the consistency of local context offers important information for robust tracking. Wen et al. [44] present a tracking algorithm based on spatial-temporal context model, which exploits temporal context to prevent drifting and spatial context to support tracking. Li et al. [45] consider spatial model consistency and temporal model coherence by jointly optimizing a spatial-temporal multi-task learning problem. Zhou et al. [46] propose a tracking framework for online learning and joint optimization of the temporal appearance model and spatial constraint model, which exploits historical information of the target and the spatial correlations between the target and the background for achieving improved performance.

Inspired by the successful application of sparse representation and context information in visual tracking, recently, several tracking methods achieve better performance by combining both of them [11,47]. Zhang et al. [11] integrate context information into representations of candidate samples by formulating object tracking task as an exclusive sparse learning problem. The representation of the best candidate is of exclusive property to context templates. Feng et al. [47] adopt sparse representation and spatial-temporal context to measure three kinds of similarities between candidates and the target object. In our work, a simple yet robust local sparse representation based tracking algorithm is developed with context information. Comparing with the methods mentioned above, we fuse spatial-temporal context information into structural local sparse appearance model rather than global sparse appearance model. Moreover, we use this discriminative information to adaptively assign varying impact factors to different local patches with consideration of local appearance difference among them.

2.4. Dictionary update in sparse representation based tracking

There is no doubt that online update of dictionary is an indispensable component in a whole sparse representation based tracking system. If effective appearance changes (e.g., pose variation) are updated into the dictionary, the representation ability of dictionary will be enhanced. On the contrary, ineffective appearance changes (e.g., occlusion) may render the tracker drift away gradually. In [15], Mei and Ling initialize the weight of each target template with its $\ell_2$ norm firstly. When the tracking result is obtained, it will be directly updated into the dictionary by replacing the least important (with the minimal weight) template. All weights are reassigned depend on the similarities between target templates and the tracking result. Jia et al. [17] combine sparse representation and subspace learning to update the template using the reconstructed image, where the template with shorter storage time has larger possibility to be updated. Zhong et al. [2] update the templates by fusing the initial target in the first frame and tracking result last stored. However, these update schemes are based on global templates, that is, all local patches inside the tracking result are either updated into the dictionary, or not. In order to capture all effective appearance changes of the target without inducing noise, a patch-based dictionary update scheme is developed in this paper.

3. Proposed tracking algorithm

In this section, we first show the construction of local object dictionary and local context dictionary and describe the structural local sparse appearance model. Then, we present a novel IAS using spatial-temporal context and analyze it specifically. Finally, a patch-based dictionary update scheme is introduced in detail.

3.1. Local dictionary construction

In this work, we construct local object dictionary and local context dictionary, respectively. Grayscale feature is adopted as visual appearance representation due to its efficiency and simplicity. Atoms of both local dictionaries and candidates need to be normalized to minimize the influence from varying illumination.

(1) Local object dictionary: Given the object region $R$, in the first frame with a bounding box $B$, we can obtain the set of initial templates $T = \{T_1, T_2, \cdots, T_n\}$ by translating $R$ to 8 directions (e.g., up, down, left, right, up left, down left, up right and down right) and converting grayscale features of each extracted target region to a vector. Where, $n$ is the number of target templates. Then $N$ overlapped local patches ($9$ in our experiments) are sampled inside each target region according to the same layout. These $n_k = N \times n$ vectorized local patches can be used to form the local object dictionary $D_o$, which can be written as the form below,

$$D_o = [d_1, \cdots, d_n, d_{n+1}, \cdots, d_{2n}, \cdots, d_{N-1}, \cdots, d_{Nn}].$$

Sub-dictionary $D_o^k = [d_{k(1)n+1}, d_{k(1)n+2}, \cdots, d_{kn}] \in \mathbb{R}^{n \times n}$ composed of the $k$-th local patch among all templates is called the $k$-th structural local object dictionary, where $k = 1, \cdots, N$, $m$ is the dimension of the patches.

(2) Local context dictionary: The context region $R_c$ is defined as an annular region which includes immediate surrounding background of $R$. Similar to $D_o$, we can obtain the local context dictionary $D_c$ via overlapped sliding windows whose center points fall in $R_c$. In this way, some local patches extracted from $R_c$ incorporate local appearance information from the context as well as the target object. The introduction of these local patches is able to prevent the tracker from drifting.

3.2. Structural local sparse appearance model

After obtaining the target candidates via particle filter, we extract overlapping local patches in them with the same layout in the target region. Under the tracking framework of local sparse representation, each patch within a candidate can be represented by solving $\ell_1$ minimization problem as follows,

$$\min_{\beta} \| y_t - D_o \beta \|_2^2 + \lambda_1 \| \beta \|_1$$

s.t. $\beta_k \geq 0$,
where \( y_i \) is the \( i \)-th vectorized local patch whose corresponding observation is obtained inside a candidate in the \( t \) frame and \( \beta_i \) denotes the corresponding sparse code. \( \lambda_i \) balances the weight between reconstruction error and sparsity.

Then, the structural reconstruction error of the \( i \)-th local patch can be calculated by

\[
\varepsilon_i = \| y_i - D^c_i \beta_i \|_2^2 \tag{3}
\]

where \( \beta_i \) is the sparse coding of the \( i \)-th local patch with respect to \( D^c_i \). Structuralization is realized by setting \( k = L \).

We compute the likelihood value of each candidate by considering all patches within it,

\[
L = \frac{1}{N} \sum_{i=1}^{N} \exp(-\varepsilon_i). \tag{4}
\]

Note that, this likelihood value is only an initial result. Different impacts from different patches have not yet been considered here. Further process will be introduced in Section 3.3 for a more reasonable likelihood function.

### 3.3. Impact allocation strategy

To make our tracker more robust against various challenges, we take different impacts of different local patches on the likelihood function into account. When a patch suffers from occlusion, it can be better represented by the local context dictionary and the state of this patch will be marked occluded. Apparently, only negative impact can this patch produce, hence we set the positive impact factor of it to be zero. For the reminder marked un-occluded, the more distinguishing from the spatial context a patch is, the greater positive impact degree it should have on the likelihood function, thus a higher positive impact factor will be allocated to it.

When the tracking result in the \( t-1 \) frame is determined, the sparse code \( z_i \) of the \( i \)-th patch with respect to the joint dictionary \( D = [D_o, D_i] \) can be obtained by solving

\[
\begin{align*}
\min_{z_i} & \| r_i - D z_i \|_2^2 + \lambda_2 \| z_i \|_1 \\
\text{s.t.} & \; z_i \geq 0,
\end{align*}
\tag{5}
\]

where \( r_i \) is the \( i \)-th vectorized patch whose corresponding observation is obtained within the tracking result in the \( t-1 \) frame and \( \lambda_2 \) is a tradeoff parameter which controls sparsity.

Then according to the characteristic of occluded patches and un-occluded patches analyzed above, the positive impact factor of each local patch can be calculated by

\[
P_i = \begin{cases} 
\exp\left(-\frac{1}{\delta_i^2 - \delta_0^2}\right), & \delta_i^2 - \delta_0^2 \geq \delta_0 \\delta_i^2 - \delta_0^2 < \delta_0, & 0
\end{cases}
\tag{6}
\]

where \( \delta_i^2 = \| r_i - D z_i \|_2^2 \) and \( \delta_0^2 = \| r_i - D^c z_i \|_2^2 \) are the reconstruction error of the \( i \)-th local patch inside the tracking result under the \( D_i \) and \( D^c_i \), respectively, \( z_i \) and \( z^c \) are the corresponding sparse codes. \( \sigma \) is the parameter that affects the degree of positive impact. \( \delta_0 \) is the threshold which determines whether the patch is occluded or not.

On the basis of the temporal context, we use the positive impact factors calculated in the previous frame to help verify candidates in the current frame. With spatial-temporal context information, we revise the likelihood function in Eq. (4) as follows,

\[
L = \frac{1}{N} \sum_{i=1}^{N} p_i^{-1} \exp(-e_i^t), \tag{7}
\]

where \( p_i^{-1} \) is the positive impact factor of the \( i \)-th patch calculated in the \( t-1 \) frame, \( e_i^t \) is the structural reconstruction error of the \( i \)-th patch obtained in the \( t \) frame. The final tracking result is the candidate with the highest likelihood value.

It is worth noting that Eq. (7) can be analyzed from two different perspectives. On the one hand, the tracking result is determined by combining all local patches inside candidates with impact allocation strategy. On the other hand, the locating of the target object depends on the collaborative work of two components. One of them is structural reconstruction error based local sparse model, while another one is spatial-temporal context based local sparse model. By combining these two components, the proposed tracker is of strong representative power as well as discriminative ability, thereby obtaining a more precise tracking result.

### 3.4. Patch-based dictionary update scheme

Local context dictionary is updated every frame to adapt appearance changes of immediate surrounding background in time over the process of tracking.

For local object dictionary, we check the \( i \)-th patch within the current tracking result using sparsity concentration index (SCI) \[23],

\[
SCI(\beta_i) = \frac{N \times \| \beta_i \|_1 / \| \beta_i \|_1 - 1}{N - 1} \in (0, 1]. \tag{8}
\]

Sparsity concentration index reflecting the concentration degree of sparse coefficients is a good tool for checking whether the patch is valid. Ideally, if \( \beta_i \) is sparse enough and concentrates on \( \beta_o \), it can be inferred that \( SCI(\beta_i^t) = 1 \), the patch may be considered perfect and can be updated. On the contrary, if \( \beta_i \) is so dense that it makes \( SCI(\beta_i^t) = 0 \), we may think the patch is much poor and abandon it. Therefore, a more valid patch will get a higher SCI score. In addition, in order to fully use the spatial context to eliminate some disturbance at the same time, the patch to be updated needs to satisfy

\[
SCI(\beta_i^t) > \theta_1 \text{ and } \delta_i^2 > \theta_2, \tag{9}
\]

where \( \theta_1 \) and \( \theta_2 \) are the thresholds which determine whether the patch is effective.

The atom with the weakest representation ability (the corresponding sparse coefficient is the minimal) in the \( i \)-th structural local object dictionary will be replaced by the \( i \)-th vectorized patch extracted from the result if this patch is effective. Through checking and updating effective patches each frame, the local object dictionary captures appearance variations of the target object without losing any contributive information.

Although both the proposed IAS and patch-based dictionary update scheme concentrate on measuring the reliability of local patches, there is some difference between them. First, IAS evaluates the positive impact degree of each patch in the stage of likelihood computation, while the patch-based dictionary update scheme validates the effectiveness of each patch in the stage of dictionary update. Second, IAS exploits historical information from the previous frame to provide some guidance, while the patch-based dictionary update scheme captures the object appearance changes from the current frame. Third, IAS uses reconstruction error to compute the positive impact factor for each local patch, while the patch-based dictionary update scheme employs sparse coding to reject outliers. In comparison with reconstruction error, sparse coding are better statistics for validation. The whole tracking process is shown in Fig. 1 and outlined in Algorithm 1.
Algorithm 1. The proposed tracking algorithm

Input:
- image sequences \( f_1, f_2, \ldots, f_M \);
- target state \( s_0 \) at the first frame;
- the number of local patches \( N \) inside an object;

Output:
- tracking results \( s_2, s_3, \ldots, s_M \);
1: construct initial local object dictionary \( D_c \);
2: construct initial local context dictionary \( D_c \);
3: initialize positive impact factors \( p_i = 1, 1 = 1, \ldots, N \);
4: for \( m = 2 \) to \( M \) do
5: generate target candidates \( X \) using particle filter.
6: compute initial likelihood value for each candidate using Eq. (4).
7: locate the tracking result \( s_m \) using Eq. (7).
8: compute positive impact factor \( p_i \) for each local patch in the tracking result \( s_m \) by Eqs. (5) and (6).
9: update local context dictionary \( D_c \).
10: for \( i = 1 \) to \( N \) do
11: replace the corresponding local patch with the weakest representative ability for updating local object dictionary if Eq. (9) satisfied.
12: end for
13: end for

4. Experimental evaluation

In this section, we illustrate the experimental methodology and conduct performance evaluation for demonstrating the effectiveness of the proposed tracking method. Our work is performed on a PC machine with Intel Core i5-5200U CPU 2.2 GHz and 8G memory. The source code is implemented in MATLAB. All experiment evaluations are based on one-pass evaluation (OPE) criterion.

4.1. Experiment setup

4.1.1. Evaluated algorithms and datasets

We compare the proposed method against several state-of-the-art trackers, including KCF [5], DSST [48], SCSD [20], CNT [6], Struck [49], TLD [1], CXT [50], SCM [2], ASLA [17], MTT [14], ISGL [13], SST [21], RSST [51], LNL [52], SALSC [53]. All these trackers are evaluated on the OTB2013 [25] benchmark with 51 challenging sequences. In the precision plots, the proposed method achieves 75.3% DP, which ranks first among all compared trackers. In the success plots, our method obtains the comparable performance with the AUC of 54.6%. Note that the proposed method, SCSD, SCM and ASLA trackers are all based on the local sparse appearance model, but our method achieves the best performance with a significant advantage. The main reason for this is that we assign different impact factors to different local patches using spatial-temporal context. It can be revealed that the IAS developed in this work is effective for robust tracking.

4.1.2. Parameter setup

The parameters involved in the proposed method keep the same during tracking for all 51 test sequences. For 9 object templates (the object in the first frame along with its translations for 8 directions) and 600 target candidates, we first resize them to \( 32 \times 32 \) pixels, and then extract local patches within them with the size of \( 16 \times 16 \) and 8 pixels as step length. The outer width and height of the context region \( R_c \) is set to twice the width and height of the object region \( R_o \). The local patches with the same size as patches in \( R_c \) are extracted in \( R_c \) with 4 pixels as step length. The parameters \( \tau_1 \) in (2), \( \tau_2 \) in (5), \( \sigma \) in (6) are set to 0.01, 0.01, 0.5. The threshold \( \delta_0 \) in (6), \( th_1 \) in (9), \( th_2 \) in (9) are set to 0, 0.5, 0.6 respectively. The 6 affine parameters for particle filter are [8,8].

4.2. Evaluation metrics

Precision plot: The precision plot shows the percentage of frames in which the center location error between the tracking result and ground-truth is less than a certain threshold. We usually concern the precision under the threshold \( s = 20 \) pixels and refer this value as DP (distance precision).

Success plot: The success plot indicates the percentage of successfully tracked frames in which the bounding box overlap between the tracking result and ground-truth is larger than a certain threshold. AUC (area under curve) of the success plot is an acknowledged evaluation metric in visual tracking.

4.3. Quantitative comparisons

4.3.1. Overall performance

Fig. 2 illustrates the overall performance of 9 compared tracking methods as well as our method via the precision plots and success plots of OPE. The overall performance is evaluated over all 51 sequences. In the precision plots, the proposed method achieves 75.3% DP, which ranks first among all compared trackers. In the success plots, our method obtains the comparable performance with the AUC of 54.6%. Note that the proposed method, SCSD, SCM and ASLA trackers are all based on the local sparse appearance model, but our method achieves the best performance with a significant advantage. The main reason for this is that we assign different impact factors to different local patches using spatial-temporal context. It can be revealed that the IAS developed in this work is effective for robust tracking.

4.3.2. Attribute-based performance

We analyze the performance of our tracker and other trackers from different aspects for further robust evaluation. As shown in Figs. 3 and 4, we draw precision plots and success plots of all evaluated trackers for each challenging attribute, respectively. Our methods ranks within top 3 on 7 out of 11 attributes in the precision plots and on 8 out of all attributes in the success plots. Generally, the proposed method performs favorably against other comparative methods in terms of deformation, background clutter, illumination variation, in-plane rotation, out-of-plane rotation, occlusion and scale variation. The improved performance can be attributed to several reasons: (1) we consider the different impacts of different local patches on the likelihood function, which makes the likelihood evaluation of candidates more accurate and robust to some challenges (e.g., occlusion, rotation); (2) it is helpful to use discriminative and historical information to handle appearance changes (e.g., background clutter); (3) patch-based dictionary update scheme can capture effective appearance changes (e.g., deformation) even in a terrible result with much occlusion.

4.4. Qualitative comparisons

We select the tracking results of several frames from nine challenging sequences for visual comparisons. These sequences are divided into three groups with three specific attributes, which are occlusion, background clutter, out-of-plane rotation. The bounding box will not appear when the corresponding tracker locates the target out of the vision. Analysis for each attribute is provided as follows.

Occlusion: The targets are easily occluded by other objects in the dynamic tracking scene as shown in Fig. 5. In the walking2
sequence, the walking woman is almost occluded completely by another walking man with similar wearing in some frames (e.g., #200). The Struck, TLD, CXT, ASLA trackers drift away gradually (e.g., #200, #325), while SCM and our method are able to track the target throughout the entire sequence (e.g., #500). In the david3 sequence, all the other trackers fail to track the target when david undergoes heavy occlusion twice with significant deformation (e.g., #85, #130, #153, #186). Only the proposed method can lock on the correct target from beginning to the end (e.g., #252). In the woman sequence, partial occlusion happens to the target frequently (e.g., #125, #293), which leads to the failure of the TLD, CXT, ASLA trackers. However, Struck, SCM trackers and our method perform well in this sequence. IAS and patch-based dictionary update scheme help our tracker to achieve better performance in terms of occlusion against other trackers.

Background clutter: As can be seen in Fig. 6, the targets in three sequences undergo background clutter. In the trellis sequence, most trackers can track the target successfully, but our method achieves the best performance compared to them. In the subway sequence, the target is interfered by other passing pedestrians.
ans many times (e.g., #42, #52, #96). TLD, ASLA, CXT trackers lose the track of the target, whereas Struck, SCM and our method still track the target without distraction. Basketball sequence is a difficult data set as many objects similar to the target exist in this sequence. All trackers drift away to wrong objects except our tracker when these objects are close to the target (e.g., #483, #660). It is because of the spatial-temporal context based local sparse model which provides discriminative and historical information to help tracking that our method is robust to background clutter.

Out-of-plane rotation: The tracking results on three sequences with out-of-plane rotation are demonstrated in Fig. 7. In the freemand sequence, we can observe that only our method tracks the man all the time, other trackers fail to track the target when it undergoes great out-of-plane rotation (e.g., #90, #220, #265). In the shaking sequence, the man always shakes his head drastically. Other methods drift away from the target during tracking (e.g., #77, #247, #350), while our method can keep track of the target persistently. Compared with the other methods, the proposed method can locate the target more accurately on the sylvester...
sequence with significant appearance changes caused by out-of-plane rotation (e.g., #1080).

4.5. Comparison with different configuration

In this section, we carry out ablation studies to test the effectiveness of each component. The proposed method mainly contains two important components, namely IAS and patch-based dictionary update scheme (readers can return to Sections 3.3 and 3.4 for more details). Therefore, we design three corresponding comparison experiments: the SLS_IAS_PDUS method, the SLS_IAS method and the SLS_PDUS method. The SLS_IAS_PDUS method represents the proposed method in this paper. The SLS_IAS method and the SLS_PDUS method exploits structural local sparse appearance model with IAS and patch-based dictionary update scheme, respectively. Fig. 8 shows the precision plots of our method with different configuration. SLS_IAS and SLS_PDUS achieve DP scores of 66.1% and 67.3%, which are both lower than the proposed

---

Fig. 4. Performance evaluation for 11 different attributes using success plots of OPE. The number of sequences for each attribute is showed in the title.
Fig. 5. Some screenshots of the tracking results in the walking2, david3, woman sequences. The targets in these sequences undergo occlusion. The frame numbers are showed with yellow font on the upper corner of the results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Fig. 6. Some screenshots of the tracking results in the *trellis*, basketball, *subway* sequences. The targets in these sequences undergo background clutter. The frame numbers are showed with yellow font on the upper corner of the results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
method with a big gap. From this result, we can conclude that IAS and the patch-based dictionary update scheme play important roles in this paper. The performance will degrade without any key component.

4.6. Parameter analysis

In this section, we show how to determine the values of several key parameters produced in this paper, such as $\delta_0$ in Eq. (6), $\eta_1$ in Eq. (9), $\eta_2$ in Eq. (9). Meanwhile, we analyze the effect of them on the OTB2013 benchmark with DP scores.

Fig. 7. Some screenshots of the tracking results in the freeman4, shaking, sylvester sequences. The targets in these sequences undergo out-of-plane rotation. The frame numbers are showed with yellow font on the upper corner of the results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Effect of $d_0$: The parameter $d_0$ in Eq. (6) is to judge whether a patch undergoes occlusion or not. When detecting an occluded patch, we will set the positive impact factor to be zero as it produces only negative impact. As for unoccluded patches, the positive impact factor will be adaptively assigned based on their ability distinguishing the context. To test the sensitivity of $d_0$, we parameterize it by a discrete set $\{-0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3\}$. As can be seen in Fig. 9, the best performance is achieved by setting $d_0$ to 0.

Effect of $th_1$: The parameter $th_1$ in Eq. (9) measures whether the patch is a valid patch from the target object. Since this parameter is closely related to the update of the object dictionary, it has significant impact on the tracking performance. Fig. 10(a) shows the results of the proposed method with different $th_1$. From the results, we can observe that too large or too small $th_1$ will lead to degraded performance. Therefore, we set $th_1$ to 0.5 because it achieves the best performance.

Effect of $th_2$: The parameter $th_2$ in Eq. (9) is a threshold to exclude the patches in the context region. When $th_2$ is small, the proposed is vulnerable to occlusion since many bad patches are taken for the object patches. When $th_2$ is large, our tracker is sensitive to deformation because the object dictionary is hard to update with a tight constraint. In Fig. 10(b), we report the DP scores of the proposed method exploiting different $th_2$. When $th_2$ is set to 0.6, the DP score of the proposed method is highest.

4.7. Comparison with sparse trackers

We evaluate the proposed method on the benchmark dataset with 10 well-performed sparse trackers, which are some most related works to ours. Note that these compared trackers are some representative sparse algorithm. Among these sparse trackers, T1, MTT, ASLA, SCM are some classical algorithms, while SCSD, ISGL, SST, RSST, LNL, SALSC are some recent popular algorithms. The results are reported in Table 1. In terms of DP scores, our method obtains performance improvements of 1.3% compared with the second best sparse tracker. Meanwhile, our method achieves competitive performance in terms of AUC scores, which is close to the
SALSC method. These results demonstrate that our method performs favorably in sparse representation-based tracking.

From the results, we can easily find that there is a big gap between existing sparse trackers and deep learning-based trackers, such as SiamFC [54] (with 81.5% DP and 61.2% AUC), FCNT [55] (with 85.6% DP and 59.9% AUC), DRT [56] (with 87.5% DP and 65.5% AUC), just to name a few. Thanks to the robust deep features, deep learning-based methods achieve an amazing tracking accuracy compared with sparse representation-based methods, which only use grayscale feature because of the computational efficiency. How to exploit richer feature representations and operate efficiently may be the goal of sparse trackers in the future.

5. Conclusion

In this paper, we propose an effective tracking method based on context-aware local sparse appearance model. With consideration of appearance variation difference among different local patches, we assign varying positive impact factors to them using spatial context, which adds some discriminative information to alleviate tracking drift. Moreover, historical information is also induced to provide support for more robust tracking with temporal context. In order to capture all effective appearance changes without missing, a patch-based dictionary update scheme is developed to improve the tracking performance. Experiment results on the recent benchmark demonstrate the better performance of the proposed method compared to other state-of-the-art methods.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant 2017YFB0202901, 2017YFB0202905). This work was partially supported by the National Natural Science Foundation of China (No.61672215, No.91320103), the Special Project on the Integration of Industry, Education and Research of Guangdong Province, China (No.2012A090300003) and the Science and Technology Planning Project of Guangdong Province, China (No.2013B090700003). The corresponding author of this paper is Mannman Peng (pengmanman@hnu.edu.cn).

References
