Occlusion detection via correlation filters for robust object tracking

Wei Zhang1,2, Baosheng Kang1

1School of Information Science and Technology, Northwest University, Xi’an 710127, People’s Republic of China
2Department of Computer Science, Baoji University of Arts and Sciences, Baoji 721016, People’s Republic of China
E-mail: zhangwei.personal@163.com

Published in The Journal of Engineering; Received on 16th October 2017; Revised on 13th March 2018; Accepted on 12th April 2018

Abstract: This study presents a robust object tracking method based on occlusion detection via correlation filters. In the proposed method, multi-feature kernelised correlation filter is employed to estimate the preliminary location of the tracked target. To predict target occlusion state, the intrinsic relationship between the most reliable tracked target and its context information is exploited via correlation filters, together with a response stability constraint to make the detection more reliable. A long-term filter is activated to recover the target if the occlusion occurs. Furthermore, the model is updated adaptively based on the changes of occlusion state and target appearance to make the tracking process robust. Extensive experimental results demonstrate that the proposed tracking method with occlusion detection performs favourably against 15 state-of-the-art trackers over 100 challenge sequences on the object tracking benchmark OTB-2015.

1 Introduction

Visual object tracking plays a crucial role in a wide range of computer vision applications, varying from video surveillance, motion recognition, to human–computer interaction and traffic control. Recently, correlation filter-based trackers (CFTs) [1–3] have received considerable attention of corresponding researchers. Employing correlation filter-based formulations has become a recent trend in tracking community, especially since Bolme et al. successfully applied a minimum output sum of squared error (MOSSE) filter [1] to tracking. By exploiting different characteristics, such as circular structure, effective feature representation, e.g. colour naming (CN) [4], and histogram of gradient (HOG) [5], and the kernel trick, varieties of works have been done to boost the overall tracking performance. Henriques et al. [2] proposed a circulant structure with kernels (CSK) tracker, which formulated tracking as a kernelised least squares classification problem. The CSK tracker was subsequently extended to multi-channel features [3]. Danelljan et al. [6] applied CN feature [4] to CSK tracker. To handle scale variation problem, a scale adaptive with multiple feature integration tracker [7] and a discriminative scale space tracker (DSST) [8] were proposed. Zhang et al. [9] proposed a novel tracking method by formulating the spatial–temporal relationships between the object and its local context in a Bayesian framework.

Despite the improved performance, these conventional CFTs still suffer from several problems in realistic scenarios. The main challenge arises from two aspects. One comes from the significant change of target appearance caused by factors such as heavy occlusion, fast motion, deformation, and motion blur. The object model should be maintained to account for such changes via online updating. Since there is no failure detection mechanism in conventional CFTs, some ‘noisy’ information can still be introduced into the object model through updating. The other is to learn the object representation from one image without any prior information. There is not sufficient information to be extracted for accurately localising the tracked target, especially when the target appearance degrades severely. The existing scene can provide useful context information, which might contribute to the tracking task. Some works have ever been done to improve the tracking performance by considering the context information. Dinh et al. [10] proposed a context tracker to deal with tracking failure caused by the emergence of regions having a similar appearance to the target. In their method, the context is exploited in two terms: distractors and supporters. Yang et al. also [11] employed the context of the tracking scene by integrating a set of auxiliary objects discovered by data mining. The target and its spatial context are considered within a network. In more recent work, Mueller et al. [12] proposed a novel framework by explicitly incorporating the global context information within CFTs to solve the drift problem.

Although substantial progress has been made and numerous object tracking methods have been proposed in recent years, visual tracking remains to be a difficult task. Heavy occlusion is one of the challenging problems frequently appearing during the tracking process. Motivated by the above observations, we propose to exploit this useful context information together with the response stability between consecutive frames to handle such problem. Actually, most tracking methods perform well when there is no occlusion or slight occlusion and the target appearance changes slowly. Once the tracked target is heavily occluded, with time increasing, inaccurate appearance accumulates and the model drift tends to happen. To deal with the occlusion problem, part-based trackers use multiple parts to model the target appearance. Even if the target is partially occluded, the remaining visible parts can still provide reliable information. Adam et al. [13] proposed a fragment-based object tracking algorithm, which utilised multiple image patches or fragments to represent the template object. Zhang et al. [14] proposed a partial occlusion handling tracking method by simultaneously matching parts in multiple frames. The tracking task is viewed as a part-based matching problem in these methods. It is often difficult to run in real time because the training and updating process are complex. Li et al. [15] and Liu et al. [16] attempted to apply the part-based strategy to correlation filter-based tracking. The part-based strategy is particularly useful in dealing with partial occlusion problems in some degree, but it still fails to track the target in some challenging scenes because it ignores the holistic information of the target. During the tracking process, the tracked target is closely related to the variation between the results in previous frames and its context information in the current frame. The surroundings of the target have a significant impact on tracking performance improvement [12]. Additionally, the target appearance should not change drastically between consecutive frames. In the previous tracking methods with occlusion detection, the context information, as well as the temporal relationship between consecutive frames, has not been made full use. Moreover, the model update, which uses a linear interpolation strategy with fixed learning rate to adapt to the appearance changes over time, is almost inevitable to
incorporate erroneous information, especially when the current tracking result is unreliable.

To address the above issues, in this paper, we present a robust object tracking method based on occlusion detection via correlation filters. Owing to the consideration of the context information together with the response stability between consecutive frames, the occlusion detection accuracy could be improved. A long-term filter is activated to recover the target if the occlusion occurs. Moreover, the model is updated adaptively based on the changes of occlusion state and target appearance to make the tracking process robust.

2 Proposed method

In this section, we give a detailed description of our method with occlusion detection. Fig. 1 shows the flowchart of our method, which is illustrated by the coke sequence in the 14th frame with occlusion. Our method contains four modules, tracking by correlation filter, occlusion detection scheme, object recovery, scale estimation, and model update, respectively. Tracking by correlation filter module is responsible for estimating the preliminary location \( I_t \). After feature extraction, the target is tracked by correlating the learned correlation filter over an image patch centred around the last target location \( I_{t-1} \). The preliminary location \( I_t \) is predicted by maximising the response map. Occlusion detection module consists of two metrics, which are evaluated by the context-aware occlusion handling scheme and the response stability constraint simultaneously. If the target is occluded, a long-term filter is activated to recover it and the current location is updated to \( I_t' \). Otherwise, the location remains unchanged. In the last module, the scale estimation is performed by constructing a scale feature pyramid, and the model is updated adaptively based on the detected occlusion state and significant appearance change.

2.1 Tracking by correlation filter

In CFTs, the appearance of the target is modelled by using a filter \( w \) trained on an image patch \( x \) of size \( M \times N \), which is \( p \) times larger than the target size to provide some context, centred at the target position. To predict the probability of the image patch being the target tracked, a classifier \( f(x_{m,n}) = \langle w, \phi(x_{m,n}) \rangle \) is trained on all circular shift of \( x_{m,n} \in [0, 1, \ldots, M-1] \times [0, 1, \ldots, N-1] \), where \( \langle \cdot, \cdot \rangle \) is dot product and \( \phi \) is the mapping to kernel space. Each shifted sample \( x_{m,n} \) corresponds to a Gaussian function label \( y(m, n) \). Then the classifier is trained by solving the minimisation problem of ridge regression:

\[
w = \arg\min_w \sum_{m,n} (\langle w, \phi(x_{m,n}) \rangle - y(m,n))^2 + \lambda \|w\|^2
\]

where \( \lambda (\lambda \geq 0) \) is the regularisation parameter that controls overfitting. With the kernel trick, the solution \( w \) can be represented by a linear combination of training samples: \( w = \sum_{m,n} \alpha(m,n) \phi(x_{m,n}) \). Employing a kernel \( \kappa(x, x') = \langle \phi(x), \phi(x') \rangle \), the filter coefficients \( \alpha \) in the Fourier domain can be computed as:

\[
F(\alpha) = \frac{F(y)}{F(\kappa(x, x)) + \lambda}\]

where \( F \) and \( F^{-1} \) denote the discrete Fourier transform (DFT) and its inverse transform, respectively. Gaussian function label is represented as \( y = [y(m,n)|(m,n) \in [0, 1, \ldots, M-1] \times [0, 1, \ldots, N-1]] \). After training process, the detection task is carried out on an image patch \( z \) in the new frame with search

---

Fig. 1 Flowchart of the proposed method

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)
where $\hat{x}$ is the learned target appearance model and $\circ$ denotes Hadamard product operator. Therefore, the new position of the target is estimated by finding the location of the maximum value of $\hat{y}$:

$$
\hat{y} = f(z) = \mathcal{F}^{-1} (\mathcal{F}(\alpha(z, \hat{x})) \circ \mathcal{F}(\alpha))
$$  

(3)

If $i = 0$, $\hat{x}_0$ and $\hat{y}_0$ represent the learned appearance model and response map of the most reliable tracked target, $z_0$ denotes an image patch with search window size $w_0 \times h_0$ cropped at the last target location in the new frame, which is illustrated in Fig. 1. $w_0$ and $h_0$ are the width and height of the target. Otherwise, $i = 1, 2, \ldots, C$, where $C$ denotes the number of context patches. $z_c$ denotes the context patches, which are sampled uniformly around the target. Fig. 1 illustrates the case when $C = 4$. Since $\hat{y}_c$ is calculated under the same appearance model with $\hat{y}_0$, if the occlusion occurs, both the response map $\hat{y}_c$ and $\mu_c$ change dramatically. Therefore, we define the context-aware occlusion factor $D_C(t)$ as the first metric:

$$
D_C(t) = \max \hat{y}_0 - \max (\hat{y}_c^t, \hat{y}_c^t, \ldots, \hat{y}_c^t)
$$  

(6)

where $\hat{y}_c$ denotes the response map of the most reliable tracked target in frame $t$. max($\cdot$) returns the maximum value. When there is occlusion, the maximum response values of the context patches and the most reliable target get close to each other. A smaller value in $D_C(t)$ indicates the likely occlusion state. Fig. 2 gives an illustration of the variation of maximum response value of the most reliable tracked target $z_0$ and those of its context patches $z_c$ in tiger1 sequence. It can be found that the proposed scheme can roughly detect the change of the target occlusion state.

### 2.2 Occlusion detection scheme

Different from existing methods [18, 19], which use a single score or overlap rate to determine the track failure or occlusion state, we design a novel occlusion detection scheme by considering the intrinsic relationship between the most reliable tracked target and its context information via correlation filters, together with the response stability between consecutive frames to make the detection more reliable. As discussed beforehand, context information of the tracked target is important for successful tracking. For example, the appearance model is trained on the image patch that consists of the target and its surrounding context, which make the model more discriminative to differentiate the target from the background. Also, once the target is occluded, it might be occluded by either the background or itself, which is closely related to its context patches. This observation can help identify the occlusion state of the target.

Therefore, we present a novel occlusion detection scheme to determine whether the target is occluded or not based on the observation that an occluder is usually more similar to at least one of its context patches. The exploitation of context information to determine occlusion state mainly includes two aspects: how to select the context patches and how to measure the similarity between the tracked target and its context patches. We use the most generic and naive method to select the context patches. They are simply sampled uniformly around the target without overlapping. The Euclidean distance metric is a simple way to measure the similarity. However, as mentioned in [20], under this distance metric, the brighter local parts would be intrinsically assigned heavier weights, which are less robust and accurate, since the brighter part does not mean it is more discriminative.

#### 2.2.1 Context-aware occlusion handling

In CFTs, the target of interest in the scene often corresponds to the correlation peak in response map, while the background shows lower responses. If the occlusion occurs, correlation filter produces a relatively lower response to the target object, which narrows the difference between its peak value and those of its context. Thus, the context-aware occlusion handling scheme can be implemented via correlation filters. We train a discriminative regression model for the most reliable tracked target $x_0$, which is illustrated in the top left image of Fig. 1 with a red rectangle. HOG descriptor is extracted as the feature. After obtaining the filter coefficients $\alpha_0$ in the Fourier domain use (2), we calculate the response map $\hat{y}_0$ of the most reliable tracked target and its context patches using a similar way to (3):

$$
\hat{y}_i = f(z_i) = \mathcal{F}^{-1} (\mathcal{F}(\alpha(z_i, \hat{x}_i)) \circ \mathcal{F}(\alpha))
$$  

(5)

A smaller $D_B(t)$ indicates that the current state tends to be more stable than the one with a larger $D_B(t)$. In contrast, a larger $D_B(t)$ means the target appearance varies significantly. After defining the context-aware occlusion handling scheme $D_C(t)$ and response stability constraint $D_B(t)$, the target occlusion state $S_O(t)$ in the $r$th frame is determined via the following equation:

$$
S_O(t) = \begin{cases} 
\text{true} & \text{if } D_C(t) < \lambda_C \text{ and } D_B(t) > \lambda_B \\
\text{false} & \text{otherwise}
\end{cases}
$$  

(10)

where $\lambda_C$ and $\lambda_B$ are the thresholds. The combination of these two metrics makes the occlusion detection more robust. If $S_O(t)$ is true,
it means that the tracked target is occluded. We activate a long-term filter to recover it.

2.3 Object recovery

Here, we briefly describe the long-term filter, which is implemented by an online CUR filter [19]. During the tracking process, all of the historical object representations with high confidence can form a large data matrix \( \mathbf{A} \). The matrix \( \mathbf{A} \) can be approximated by the online CUR to represent the intrinsic object structure. We generate the online CUR filter \( \mathbf{h}_{\text{cur}} \) by sampling \( c \) columns of matrix \( \mathbf{A} \) randomly to form the column matrix \( \mathbf{C} \), and then average it as:

\[
\mathbf{h}_{\text{cur}} = \frac{1}{c} \sum_{i=1}^{c} \mathbf{C}(, i)
\]  (11)

After obtaining the long-term filter \( \mathbf{h}_{\text{cur}} \), we detect the most reliable target and those of its context patches in tiger1 sequence. (Red dot rectangles in the lower row indicate the likely occlusion state and the black dot rectangles indicate the non-occlusion state. Images in the upper row show the video frames sampled from these dot rectangles, respectively. The yellow #number on the top left denotes the frame index)

2.4 Adaptive model update

To track the target robustly, the target appearance model should be continuously updated as it changes over time. A linear interpolation strategy is exploited to adapt to the target’s appearance changes, which is performed on the target appearance model \( \hat{x} \) and the filter coefficients \( \alpha \) in the Fourier domain as follows:

\[
\hat{x}' = (1 - \eta)\hat{x}^{t-1} + \eta\hat{x}
\]  (14)

\[
\mathcal{F}(\alpha') = (1 - \eta)\mathcal{F}(\alpha^{t-1}) + \eta\mathcal{F}(\alpha)
\]  (15)

where \( t \) is the frame index and \( \eta \) denotes the learning rate. In many conventional CFTs, such as MOSSE [1] and KCF [3], the learning rate has a fixed value. However, during the tracking process, the tracked target often suffers from occlusion, and its appearance may change significantly. The learning rate in methods [1, 3] cannot vary according to these changes and the challenging factors exist in real scenarios. A fixed coefficient updating also increases the risk of drifting in such cases. Different from existing methods [1, 3], we introduce an adaptive way to update the model based on the changes of occlusion state \( S_{\text{oc}}(t) \) (described in Section 2.2) and target appearance. In method [1], the peak to sidelobe ratio are used to measure the peak strength and detect occlusion or track failure. However, this measurement only accounts for the maximum value variation in the current frame, it lacks consideration of the variation in temporal domain. In our method, we take the maximum response variation in the consecutive frames into consideration to reflect the appearance changes. In general, the changes of target appearance can be reflected by the correlation peak of the response maps in some degree. Larger fluctuation of correlation peaks of the response maps in the consecutive frames indicates significant appearance changes of the tracked target. We define the first indicator as follows to evaluate this fluctuation:

\[
G(t) = \left| 1 - \frac{\max(y^t)}{\max(y^{t-1})} \right|
\]  (16)
where \( \hat{y} \) denotes the response map obtained in the \( t \)th frame. Smooth variation of the target appearance between consecutive frames indicates more reliable tracking results, the ratio is close to 1, and the value of indicator \( G(t) \) approaches to 0. Otherwise, the value of \( G(t) \) will increase. Larger value of \( G(t) \) reflects significant changes in target appearance.

When the target object is occluded, some noisy information will be introduced in the appearance model and the classifier, with time increasing, eventually lead to drift. This can be alleviated by employing the target occlusion state \( S_t(t) \) as the second indicator to compensate for the updating. The method in [3] cannot handle the occlusion problem while the occlusion detection scheme in our method considers continuities in both spatial and temporal domains to make the detection process robust. Therefore, the learning rate of the correlation filter can be adjusted by considering the two indicators above simultaneously:

\[
\eta = \begin{cases} 
\beta \times \eta_{init} & \text{if } S_t(t) = \text{true or } G(t) > \lambda_G \\
\eta_{init} & \text{otherwise}
\end{cases}
\] (17)

where the parameter \( \eta_{init} \) denotes the initial learning rate, and \( \beta \) is the relative ratio to reduce the initial learning rate if the current tracking result is unreliable. \( \lambda_G \) is the threshold. For the most reliable tracked target appearance model \( x_o \) and its correlation coefficients \( \omega_o \), we just update the model when \( \max(h_o) > \lambda_o \). The scale filter \( h_i \) is updated when the tracked target is not occluded.

3 Experiments and analysis

To evaluate the performance, we conduct extensive experiments on a recent benchmark data set OTB-2015, which is composed of 100 challenge sequences. Both quantitative and qualitative comparisons are provided with 15 state-of-the-art trackers. These trackers can be broadly categorised into three classes: (i) CFTs including CSK [2], KCF [3], CN [6], DSST [8], and CCT [19], (ii) representative trackers using single or multiple classifiers, including the OAB [22], MIL [23], Struck [24], TLD [25], VTD [26], SCM [27], and TGPR [28], and (iii) context tracker CXT [10] and part-based tracers, including Frag [13] and RPT [15]. The overall experiments are conducted in Matlab 2013b on an Intel® Core™ i7-4702 CPU (2.2 GHz) with 8 GB RAM.

3.1 Experimental set-up

3.1.1 Parameter settings: In the correlation tracking part, the cell size of HOG is \( 4 \times 4 \), and the orientation bin number of HOG is 9. The parameter \( p = 2.5 \) and the initial learning rate \( \eta_{init} = 0.015 \). The Gaussian kernel function with kernel sigma 0.5 is chosen according to [3]. The parameters in scale estimation are set the same as the DSST [8]. In occlusion detection part, the number of context patches is 4. Parameters \( \lambda_c = 0.1 \) and \( \lambda_r = 0.16 \) are empirically set to a constant. Parameters in object recovery are set \( c = 20 \) and \( k = 10 \). We empirically set parameters in adaptive updating part as \( \beta = 0.1 \), \( \lambda_c = 0.4 \), and \( \lambda_r = 0.38 \). All parameters are the same for all following experiments.

3.1.2 Evaluation methodology: In the overall experiments, our method and the state-of-the-art trackers are compared by using the evaluation methodology provided by the recent benchmark data set [7]. One-pass evaluation (OPE) is employed and two metrics, precision and success plots, are used. The precision metric computes the percentage of frames in the sequence whose estimated target centre is within some certain distance with the ground truth. The average Euclidean distance between the estimated target centre and the ground truth is also defined as centre location error (CLE). Smaller CLE value means a more accurate result. The Pascal visual object classes (VOC) overlap ratio (VOR) is computed as

\[
\text{VOR} = \frac{\text{Area}(B_t \cap B_l)}{\text{Area}(B_t \cup B_l)}
\] (18)

where \( B_t \) represents the tracking bounding box and \( B_l \) the ground truth bounding box. \( \cap \) and \( \cup \) represent the intersection and union operators. Area() represents the area region. The success metric computes the overlap ratio between the tracking bounding box and the ground truth. In precision plots, the average distance precision is plotted over a range of thresholds, and the average precision score at 20 pixels threshold corresponding to the OPE of each tracker is contained in the legend. Likewise, in the success plots, the average overlap precision is plotted, and the area under the curve (AUC) is reported in the legend.

3.2 Quantitative evaluations

Table 1 shows the comparison between our method and 15 state-of-the-art trackers on the OTB-2015 data set using the median CLE and median VOR. We also report the speed in median frames per second (FPS). The best three results are illustrated in bold, underlined, and italic fonts, respectively. From Table 1, we find that our proposed method achieves the best performance regarding both median CLE and median VOR. Our method significantly improves the KCF with a relative reduction in the median CLE by 37.4%. Moreover, our method achieves 0.66 in the median VOR, which gets a 29.4% improvement upon KCF on the data set. The median speed of our method over 100 sequences is 15.5 FPS, which is about three times faster than the part-based trackers [13, 15].
also provide the overall and attribute-based performance regarding success plots in Figs. 3 and 4.

3.2.1 Overall performance: As it can be seen from the plots in Fig. 3, our method achieves 0.564 success score and 0.767 precision score, both of which rank in the first place among all the compared trackers. Our method improves the success score and precision score by 18.7 and 10.8%, respectively, compared to KCF. Moreover, our method outperforms CCT by 2.9 and 4.1%, and RPT by 6.6 and 2.8%, in success score (AUC) and distance precision at a threshold of 20 pixels.

3.2.2 Attribute-based performance: The benchmark sequences are annotated with 11 different attributes, namely: scale variation, out-of-plane rotation, in-plane rotation, occlusion, deformation, fast motion, illumination variation, background clutter, motion blur, out-of-view, and low resolution. These attributes affect the performance of a tracker and are used to evaluate the tracker in different scenarios. Fig. 4 shows the success plots of the attributes above, respectively. We mainly analyse the ranked results based on the success plots, which are more accurate than precision plot, as described in [6].

In success plots, our method ranks first on 9 out of 11 attributes, except for background clutter and low resolution. On background clutter and low resolution subset, RPT and SCM perform best. In detail, our method improves the AUC score of 11 attributes by 31.2, 17.6, 14.0, 23.3, 16.7, 15.7, 23.6, 11.1, 26.1, 29.8, and 35.5%, respectively, compared to KCF. The rank sequence is corresponding to the attribute sequence in Fig. 4, which is arranged by the number of videos associated with each attribute in descending order. Among all the attributes, scale variation, low resolution, and out-of-view performance are improved significantly. In occlusion subset, our method performs well with an improvement by 5.7 and 13.7% compared to CCT and RPT, which indicates that the occlusion detection scheme in our method is effective. Although our method is not specially designed for motion blur and illumination variation, the proposed method obtains very appealing performance on these challenging sequences.

3.3 Qualitative evaluations
To better illustrate the performance, we further present representative tracking results obtained by top ten trackers in the overall performance evaluation in Fig. 5 with several challenging sequences. Meanwhile, we also provide a frame-by-frame comparison of the centre location error (in pixels) between our method and existing trackers in Fig. 6. It can be observed from Figs. 5 and 6 that our method provides promising results compared to the state-of-the-art trackers on these sequences.

3.3.1 Deformation and pose variation: The target in basketball sequence undergoes pose variation, occlusion, and background clutter. In this case, the CCT, DSST, KCF, TGPR, CN, RPT, and our method perform reliably throughout the entire sequence, while others drift away to the background at different time points. The blurbox sequence contains pose variation, motion blur, and fast motion. In this sequence, the Struck, TGPR, RPT, and our method track the target accurately throughout the sequence, while others drift away gradually due to motion blur and fast motion. In particular, the SCM, KCF, TLD, CN, DSST, and CCT drift at frame 4, 39, 53, 196, 216, and 223, respectively. In gym sequence, except for KCF, CN, and SCM, other methods are able to track the target for the entire sequence. In panda sequence, only Struck, TGPR, SCM, and our method can successfully track the target throughout the sequence, while most of the other tracker drift away at different time points. For example, RPT drifts at frame 160. The KCF, DSST, CN, and CCT drift at about frame 320. TLD drifts at frame 617. It can be implied from all of the results that our method can handle the deformation and pose variation well by exploiting multiple features and the context information simultaneously.

3.3.2 Occlusion: As it is illustrated in Fig. 5, for the woman, box, jogging2, and lemming sequences, the target objects undergo partial occlusion or complete occlusion. In woman sequence, the target lasts serious partial occlusion for a long time. Only Struck and our method perform well for the entire sequence. The TLD, CN, and CCT drift away at frame 110, 149, and 341. The KCF, RPT, DSST, SCM, and TGPR lose the target at last since frame 561. In box sequence, the target contains partial occlusion and scale variation in a cluttered background. Except SCM drifts at frame 229, most of the trackers drift when the target is partially occluded at frame 461, but our method and TLD relocate the target since it appears at frames 485 and 494. In the jogging2 sequence, only TGPR and our method work well across the entire sequence. Other trackers lose the target after the target is completely occluded by the telegraph pole. The TLD and CCT
relocate the target after a long time. For *lemming* sequence, the CCT, Struck, TLD, and our method can recover the target since it is completely occluded at frame 346. The SCM, Struck, CCT, TLD, TGPR, CN, and DSST also exhibit drifting in the process. All the above results indicate that the occlusion detection and adaptive model update scheme in our method work together to handle the occlusion problem well in both partial and complete cases.

### 3.3.3 Illumination and scale variation:

The *car4*, *human7*, *shaking*, and *singer1* sequences contain illumination and scale variation at the same time. In *car4* sequence, the TLD and CN show a significant deviation. The CN, KCF, Struck, TGPR, and RPT cannot adapt to the scale change during the tracking process. Other trackers and our method can track the target well and adapt to the scale variation. In *human7* sequence, the target still suffers from deformation, motion blur, and fast motion. The TLD, Struck, RPT, and our method can track the target across the process, but the remaining trackers drift away at different time points, e.g. the SCM, CN, and DSST at frame 81, and KCF, CCT, and TGPR at frame 119. For *shaking* sequence, the KCF and CCT fail to track the target at the beginning due to illumination variation. The Struck and TLD also exhibit drifting, but others can track the target accurately. In *singer1* sequence, only TGPR and CCT experience drift in the process. The Struck, CN, and KCF lack the ability to cope with scale variation. Our method achieves better results on these sequences and covers the target more accurately in the presence of illumination and scale changes. This is because our method not only utilises multiple features and

---

**Fig. 4** Success plots of different attributes over 100 sequences on the OTB-2015 data set. The values appearing in the title denote the number of videos associated with the respective attribute.
context information around the target, but also updates the model adaptively to reduce the learning rate of scale model when the result is not reliable.

3.3.4 In-plane rotation and out-of-plane rotation: The targets in bolt, rubik, tiger1, and tiger2 sequences undergo significant appearance change caused by in-plane rotation and out-of-plane rotation. In addition, the tiger1 and tiger2 sequences contain occlusion, motion blur, and fast motion, the target objects in bolt and rubik sequences undergo deformation and scale variation, respectively, which make tracking more difficult. In bolt sequence, the TGPR, SCM, Struck, and RPT drift away to the background at the beginning. The TLD drifts gradually, but our method performs well throughout the whole video. For the target in rubik sequence, the RPT, SCM, and TGPR show a large deviation in the process. The Struck and CN also exhibit drifting in the process, and the CCT cannot adapt to the scale change accurately. For tiger1 sequence, the CCT, CN, RPT, and our method can track the target well, but the remaining trackers show drifting at different time instances. In tiger2 sequence, the SCM, DSST, KCF, TLD, and Struck show a large scale of deviation, the CCT, TGPR, and CN exhibit drifting at the end of the process after occlusion and out-of-plane rotation. Only RPT and our method work well for the whole sequence. All these results indicate that our method performs well in handling large appearance change caused by in-plane rotation and out-of-plane rotation. Exploiting context information of the target and its background can improve the tracking performance and achieve better results.

Fig. 5 Visualisation of the tracking results of top 10 trackers in the overall performance evaluation on 16 challenging sequences
4 Conclusions
In this paper, we propose a robust object tracking method based on occlusion detection via correlation filters. Our method takes full advantage of the context information of the most reliable tracked target to predict the target occlusion state. Meanwhile, a response stability constraint is introduced to make the detection more reliable. They cooperate well with each other to reflect both the spatial and temporal continuities. Based on the detected occlusion state, a long-term filter is utilised to recover the tracked target, which is effective and efficient. Besides, the model is updated adaptively based on the changes of occlusion state and target appearance to make the tracking process robust. The extensive experimental results demonstrate that the proposed method outperforms several state-of-the-art trackers in terms of accuracy and robustness.

5 Acknowledgment
This work is supported by the National Natural Science Foundation of China (Grant Nos: 61772421 and 61572400).

6 References

Fig. 6 Frame-by-frame comparison of the centre location error (in pixels) among top 10 trackers in the overall performance evaluation on 16 challenging sequences

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)


This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)