Li et al. EURASIP Journal on Wireless Communications and Networking (2019) 2019:261
https://doi.org/10.1186/s13638-019-1579-x

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Object tracking framework with Siamese network and re-detection mechanism

Daqun Li[1,2], Yi Yu[1*] and Xiaolin Chen[1]

Abstract

To improve the deficient tracking ability of fully-convolutional Siamese networks (SiamFC) in complex scenes, an object tracking framework with Siamese network and re-detection mechanism (Siam-RM) is proposed. The mechanism adopts the Siamese instance search tracker (SINT) as the re-detection network. When multiple peaks appear on the response map of SiamFC, a more accurate re-detection network can re-determine the location of the object. Meanwhile, for the sake of adapting to various changes in appearance of the object, this paper employs a generative model to construct the templates of SiamFC. Furthermore, a method of template updating with high confidence is also used to prevent the template from being contaminated. Objective evaluation on the popular online tracking benchmark (OTB) shows that the tracking accuracy and the success rate of the proposed framework can reach 79.8% and 63.8%, respectively. Compared to SiamFC, the results of several representative video sequences demonstrate that our framework has higher accuracy and robustness in scenes with fast motion, occlusion, background clutter, and illumination variations.

Keywords: Object tracking, Siamese network, Re-detection mechanism, Generative model, High-confidence update

1 Introduction

Object tracking is a critical issue in the field of computer vision, which exits in a wide area of applications including video surveillance, human-computer interaction, intelligent traffic monitoring, and the military, to name a few. After the arbitrary object in the first frame of the video sequence is given, how to precisely locate its position in the subsequent frames is the central problem of object tracking. Although many researchers have made great efforts to improve the tracking algorithm [1–5], the problem is still very challenging in many tracking tasks due to the factors such as occlusion, illumination variations, scale change, motion blur, and deformation. Therefore, designing an accurate and robust framework for object tracking has important value in theory and practice.

In traditional tracking frameworks, object features are manually defined or combined [6–11]. Most frameworks for object tracking can be divided into generative methods [12–15] and discriminative methods [16–25]. Although these hand-crafted features show satisfactory results, they are designed for specialized scenes and cannot perform well in some challenging conditions. For instance, generating features by pixel comparisons [25] may become inaccurate when the illumination of the background or the scale of the object changes, which requires adaptive learning methods that can capture effective changes in object appearance over time.

Recently, convolutional neural networks (CNNs) have obtained great attention as a result of their achievements on automatic feature extraction. They can learn to extract features from substantial annotated image data and a great deal of object classes. These features are rich in high-level semantic information and are adept in distinguishing different categories of objects. Motivated by this progress, several trackers (e.g., ECO [26], C-COT [27], DeepSRDCF [28], and HDT [29]) integrate CNN deep features into the conventional tracking frameworks and benefit a lot from their expressive power. Some others (e.g., BranchOut [30], TCNN [31], MDNet [32], and STCT [33]) utilize CNNs directly as a classifier and make the best of the end-to-end training. Most trackers use online training to improve the tracking performance. However, due to the great quantity of deep features and

* Correspondence: 13756006195@139.com
[1]Chinese Academy of Sciences, Changchun Institute of Optics Fine Mechanics and Physics, 3888 Dongnanhu Road, Changchun 130033, China
Full list of author information is available at the end of the article

the complexity of deep neural networks, the computational cost of online training is high. Accordingly, most CNN-based trackers run much slower than the real-time trackers.

Several recent trackers differ from the single network structure in that they utilize a Siamese network structure as the core of the framework. This method focuses on combining two CNN-based real-time trackers, which completely avoids online training and achieves high tracking speed. GOTURN [34] utilizes Siamese network as the feature extractor and the fully connected layers are adopted as the fusion tensor. SiamFC [35] removes the fully connected layer to reduce the amount of computation, and only five fully convolutional layers are applied to train an end-to-end Siamese network for learning a similarity function. Moreover, the function is directly used for online tracking without any complex fine-tuning tactics. The fully convolutional structure allows SiamFC to take full advantage of the offline training data and make itself more discriminative. Recently, there are also many following-up studies of SiamFC [36–39]. However, since the target marked with the first frame is used as a template for tracking, it is difficult to track the target accurately when the target changes. If the target changes greatly, the tracker based on Siamese network will not be able to effectively identify the target. Moreover, due to the lack of re-detection mechanism, complex background interference will cause a significant decline in tracking accuracy. Thus, there is still a gap between SiamFC and the best online tracker in tracking performance.

To achieve long-term tracking, the re-detection mechanism is often used to identify a reliable result when interference appears in the tracking process. In addition, the re-detection mechanism receives more and more attention and is closely related to the practical applications, such as multi-object tracking [40–42] and person re-identification [43, 44]. In [45], a re-detection mechanism with self-paced learning scheme is utilized to avoid contaminating the training set. The mechanism can select trustworthy frames to learn the appearance of the object. In [46], a re-detection approach is proposed to handle the problems of severe occlusion and changes in motion. The approach can deduce occlusion and changes in motion. Thus, the tracking effect can be improved to a certain extent. The multi-store tracker (MUSTer) [23] adopts a short-term method to detect the interference and a long-term method to control the output. In [47], a re-detection module is utilized to achieve long-term tracking. The module can re-detect the potential contaminated results of the classifier and decide whether to replace the original target template. Although the above methods improve the tracking effect, the re-detection mechanism needs to evaluate the output in every frame while the features in the mechanism are hand-crafted.

Motivated by the above observations, this paper aims to improve the deficient tracking ability of SiamFC in complex scenes with fast motion and similar interference. A re-detection mechanism is utilized and it adopts the Siamese instance search tracker (SINT) [36] as the re-detection network. When multiple peaks appear on the response map of SiamFC, a more accurate re-detection network can re-determine the position of the target. Meanwhile, for the sake of adapting to various changes in the appearance of the object, this paper employs a generative model based on Gaussian mixture model to construct the template of SiamFC. Furthermore, a method of template updating with high confidence is also used to prevent the template from being contaminated.

To summarize, the main contributions of this paper are listed as follows:

1. A re-detection mechanism based on Siamese network structure is proposed to improve the tracking performance. This enables us to gain a very accurate and robust result while multiple peaks appearing on the tracking response map.
2. A generative model based on Gaussian mixture model (GMM) is exploited for construction of the template, instead of a unique and fixed template in the first frame. Thus, a reliable tracking process can be achieved while tracking in complex scenes.
3. The strategy of template updating with high confidence is utilized to promote the quality of the template. This makes the framework more adaptive to various changes in appearance and similar interference.
4. The evaluation on the widely used object tracking benchmark (OTB) demonstrates that our framework has strong accuracy and robustness in complex scenes with fast motion, severe occlusion, background clutter, and illumination variations. In addition, the framework can still maintain a relatively faster tracking speed, and the running frame rate on GPU can reach 21 frames per second (fps).

The rest of the paper is organized as follows: In Section 2, we introduce the related works in details and show the flowchart in Fig. 1. The SiamFC network with re-detection mechanism as well as the construction and the updating of templates are presented in Section 3. In Section 4, the experimental details and the objective evaluation of our framework will be discussed. In Section 5, we reach the conclusions of the paper.

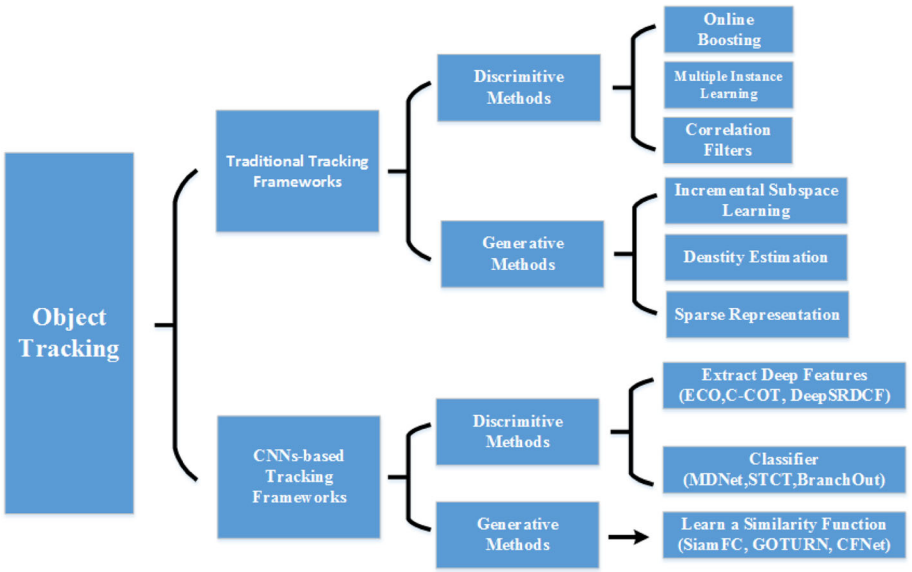

Fig. 1 The logical flowchart of the object tracking. The logical flowchart is used to clearly introduce the development of the object tracking mechanism

2 Related works

2.1 Traditional tracking frameworks

Most traditional tracking frameworks can be divided into generative methods and discriminative methods. Generative methods can depict the appearance of the object by utilizing generative models. The region which best matches the generative models will be regarded as the target region. In prior tracking frameworks, various generative methods have been proposed including incremental subspace learning [12, 13], sparse representation [14], density estimation [15], etc. The famous algorithm for incremental visual tracking (IVT) is proposed in [12]. The algorithm adopts the particle filter as the motion model and the incremental subspace to depict the appearance of the object. In [13], 2D principal component analysis has been integrated into the generative models. The sparse representation is first proposed in [14]. In this method, a set of trivial templates and a particle filter are utilized to overcome the challenging issues. At the same time, the sparsity can be obtained by solving an l_1-regularized least squares problem.

In contrast, the discriminative methods mainly learn classifiers which can distinguish the object from the complex background. Various discriminative methods have been proposed based on multiple instance learning [16, 17], online boosting [18, 19], correlation filters [20–23], etc. In [16], multiple instance learning is adopted as a learning strategy which can replace the traditional supervised learning to filter weak classifiers. In [18], online boosting is used to select features and make the visual tracking more robust. AdaBoost is proposed in [19], which combines sets of weak classifiers into a stronger one. In recent years, discriminative methods based on correlation filters have gained great attention. In [20], the minimum output sum of squared error (MOSSE) has been added into a correlation filer, which can run in hundreds of frames per second. A kernelized correlation filter (KCF) is proposed in [21]. In KCF, the circulant matrices and multi-channel features in a Fourier domain are used to improve the tracking effect. At the same time, several subsequent research works of KCF have been investigated. DSST [22] trains some separate correlation filters to overcome scaling or translation in the tracking process. MUSTer [23] adopts a short-term method to detect the interference and a long-term method to control the output. In order to weaken the response of the filter near the edge of the image and increase the response of the filter at the center of the image, SRDCF [24] introduces a spatial regularization component to the learning process.

Although these approaches show satisfactory results by using hand-crafted features, they are designed for specialized scenes and cannot perform well in some challenging conditions.

2.2 CNNs-based tracking frameworks

Recently, convolutional neural networks (CNNs) have obtained great attention as a result of their achievements in automatic feature extraction. They can learn to extract features from substantial annotated image data and a great deal of object classes. These features are rich in high-level semantic information and are adept in distinguishing different categories of objects. Similar with the

object tracking algorithms based on hand-crafted features, the CNNs-based tracking frameworks can also be divided into discriminative methods and generative methods.

2.2.1 Discriminative methods

To develop a discriminative method, a common way is to extract the deep features by using CNNs model and replace the hand-crafted features in traditional tracking framework, such as correlation filter [26–28]. In [26], deep features are combined with the hand-crafted features so that the redundancy of the algorithm can be reduced. In [27], the framework takes a joint learning strategy to fuse the deep features from different layers of the spatial pyramids. Thus, the precision of the framework can be greatly improved. In [28], the hand-crafted features in SRDCF [24] are replaced by the deep features, and the framework also suggests that the features from the first layer provide more accurate tracking performance.

In addition, CNNs can also be utilized as a classifier [30–32]. In [30], the framework combines convolutional layers with multiple branches of fully connected layers, and randomly selects a subset of branches as the regularization. In [31], the framework adopts offline pre-training to reduce the computational cost and combines the layers with a tree structure to get a robust classifier. Thus, the tracking performance can be improved. In [32], the method utilizes a large number of tracking datasets with ground truths to pre-train the CNNs model. The model which is constructed by the shared layers and the domain-specific layers can generate a generic representation of the object. Thus, the framework can effectively distinguish the object from the complex background by using the CNNs model.

Although these discriminative methods can greatly improve the tracking performance, the great quantity of deep features and the complex online fine-tuning make the framework time-consuming. Accordingly, most discriminative methods run much slower than real-time trackers.

2.2.2 Generative methods

Visual object tracking can be treated as the process of learning a similarity function. By comparing the template patch with the candidates in the search region, the object can be tracked to the position with the highest similarity score. A significant advantage of the generative method is that it requires no or little online training. Therefore, it is easy to achieve real-time tracking.

A Siamese structure can be utilized to learn a robust similarity function based on deep CNNs. As a representative of pioneering algorithms, SiamFC adopts fully convolutional layers to train an end-to-end Siamese network for learning a similarity function. The fully convolutional structure can provide a larger search image and use the image as an input to the network, rather than a candidate patch of the same size as the template patch. The structure will calculate the similarity of all sub-windows on the dense grid in one evaluation. Moreover, each sub-window can availably represent a useful sample at training time without too much additional cost.

Recently, there are many following-up studies of SiamFC [36–39]. SINT [36] adopts Siamese network to extract hierarchical convolutional features of the template patch and the search patch, respectively. Candidates are randomly cropped from the search patch, each of which is selected and compared with the search patch to acquire the best one. Optical flow is also utilized to achieve the better performance and the higher accuracy. However, because of the extensive computation of the SINT, the running speed is only 4 fps. The tracker takes no appropriate mechanism to re-detect the target and exclude the interference while tracking in scenes with occlusion or similar interference. The EAST [37] utilizes the Siamese network to efficiently extract the deep features. If the low-level features are useful enough, the tracker will attempt to stop the feature being extracted ahead of time in order to speed up. The key to this decision-making behavior is to train an agent by using reinforcement learning. Although the adaptive mechanism can decrease the computational complexity of the algorithm, the tracking effects in complex scenes have not been improved much. By online updating the embeddings of the tracked target, DSiam [38] achieves better tracking performance without too much speed loss. Moreover, this method proposes a dynamic Siamese network and utilizes the previous frames to learn the changes in appearance of the target. However, the similar interference, occlusion, and fast motion are still the key issues affecting the tracking performance of this tracker. CFNet [39] integrates the correlation filters into the template branch, which makes the Siamese network shallower and more efficient. This not only improves the tracking performance of the network while using the shallow structure, but also combines the deep features with the correlation filter perfectly. Nevertheless, without rational strategy of model updating, the tracker cannot achieve satisfactory effects in some complex scenes.

Inspired by the above research works, this paper inherits the network structure from SiamFC. A fully-convolutional Siamese network is leveraged as the main tracking algorithm. Meanwhile, a re-detection mechanism as well as the construction and the updating of templates are also added to improve the tracking performance of our framework.

2.3 Template-based tracking

In order to improve the tracking accuracy of the tracker in complex scenes, some trackers introduce template matching into the framework. The tracker based on template matching directly compares the template patch with the object patches sampled from the current frame. Normalized cross-correlation (NCC) is a classical method that is often used in matching algorithm. It can calculate the correlation between two sets of sample data and the range of the values is between − 1 and 1. The whole image or the region of interest can be regarded as a set of sample data (search region) in object tracking. The higher the correlation between a subset (template image) and the sample data is, the closer the result is to 1. Instead, the closer the result is to − 1. The result can be obtained from the following formula.

$$R(i,j) = \frac{\sum_{a=1}^{M}\sum_{b=1}^{M} S(a,b) \times T(a,b)}{\sqrt{\sum_{a=1}^{M}\sum_{b=1}^{M} [S(a,b)]^2}\sqrt{\sum_{a=1}^{M}\sum_{b=1}^{M} [T(a,b)]^2}} \quad (1)$$

Here, $T(a,b)$ is the template image, $S(a,b)$ is the search region of the target, M is the height and the width of the data. Because of its simplicity for implementation, NCC has been utilized in some recent trackers such as TLD [25]. However, when the illumination of the image has changed or the target has deformation, NCC usually has low sensitivity and the similarity between the target and the template will decrease. Oron et al. [48] proposes another representative template-based tracker, which improves the Lucas-Kanade tracking framework, and unites the pixel-wise segregation of background/object with the template matching to construct a unified Bayesian framework. Bertinetto et al. [35] and Tao et al. [36] focus on using CNN model to learn a robust and accurate similarity function, which can efficiently match the certain template within a search region.

Since the appearance of objects always suffers challenging situations, the template should be updated online. Online template updating is first proposed in [49], where three factors, namely, the noise, the stability and the transient factor, are exploited to represent the object. Other instances of such tracking method are listed in [12, 50]. For maintaining the robustness of the appearance model, [50] proposes a novel strategy of template updating. While [12] tries to update the appearance-based subspace online through incremental PCA learning. Recently, incremental 2DPCA representation has been introduced to target objects [13]. Because of the nature of the 2D image, this kind of method shows a promising tracking performance. In fact, by continuous or intermittent online updating, these methods can obtain a compact representation of the object. However, they are prone to drift since the templates are contaminated. Moreover, the templates always have limited modeling abilities because they only represent a single appearance of the object. When the appearance of the object changes greatly, the tracker will fail to track it.

Inspired by the previous research works, a generative model has been used in our tracking framework to cope with the variations in appearance, and a strategy of template updating with high confidence is utilized to avoid the drift of the bounding box in the tracking process.

3 Methods

In this section, we will introduce the proposed tracking framework in detail, which is abbreviated into Siam-RM. We first introduce the SiamFC network, which is improved by the re-detection mechanism. Secondly, the generative model for constructing templates will be described. Finally, a strategy of template updating with high confidence will be introduced. Figure 2 presents a detailed flow chart of the Siam-RM framework.

3.1 Improved SiamFC

3.1.1 Fully-convolutional Siamese networks

Fully convolutional Siamese networks (SiamFC) adopt fully convolutional layers to train an end-to-end Siamese network for learning a similarity function. The network is shown in Fig. 3 and it uses a template patch (T) and a larger search region (S) as inputs. The embedding function of the Siamese network is a feature extractorF_ϕ, which extracts the features of the template and the search region at the same time. Then, the network feeds the extracted features into the cross-correlation function:

$$g_\phi(T,S) = F_\phi(T) * F_\phi(S) + b \quad (2)$$

The maximum value in final response map (left-hand side of Eq. 2) corresponds to the location of the target.

Generally, in order to preserve the real-time performance of the framework, the embedding function F_ϕ usually adopts simple network structure (e.g., AlexNet in [51]). However, the size of the response map is relatively small and is not suitable for precise positioning. In order to obtain a larger response map, we apply Baseline-conv5 model as the embedding function. The architecture is shown in Table 1.

The model is an improved version of SiamFC which is first introduced in [52]. Firstly, the model utilizes fewer steps (4 strides in total, not 8 strides) in the network to obtain a larger feature map. The feature map is beneficial to pinpoint and re-detect the location of the object.

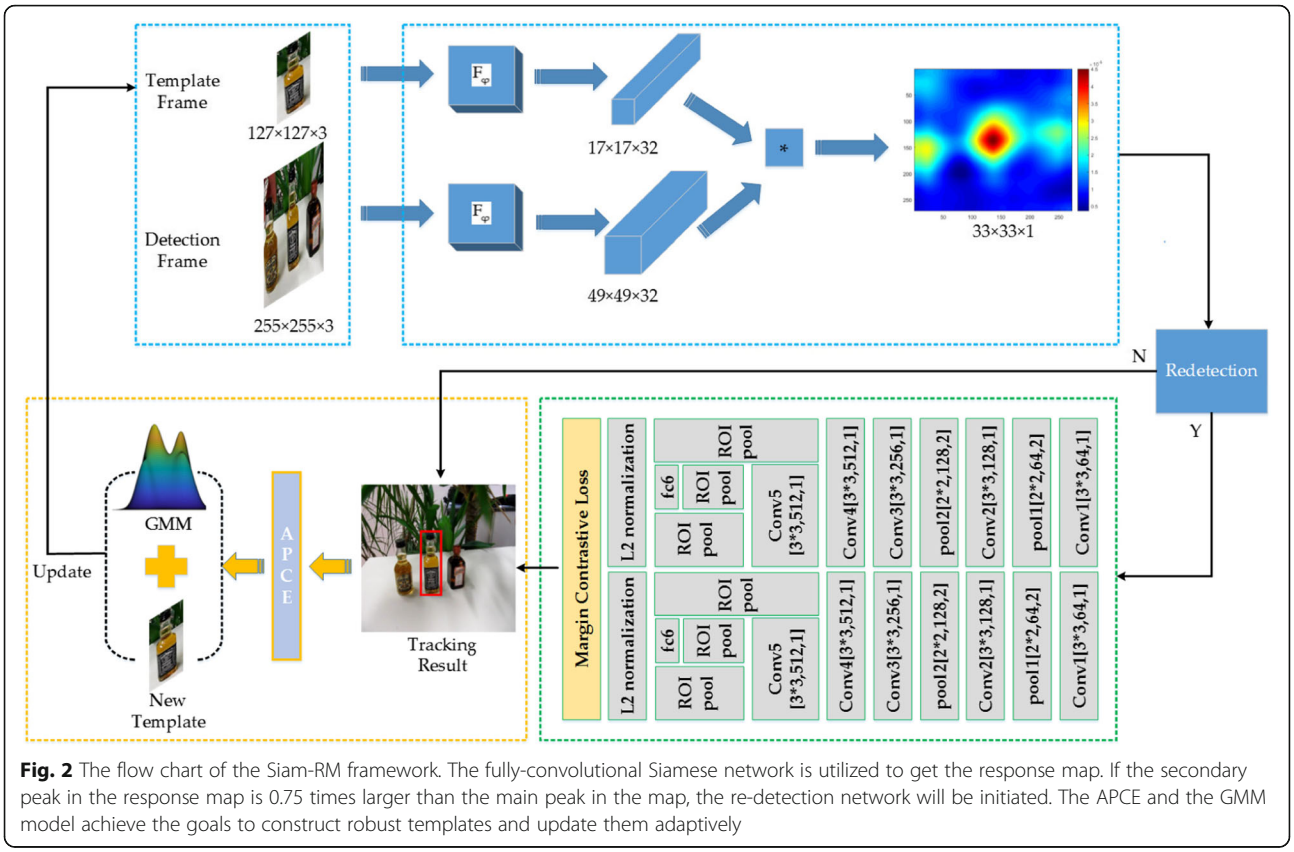

Fig. 2 The flow chart of the Siam-RM framework. The fully-convolutional Siamese network is utilized to get the response map. If the secondary peak in the response map is 0.75 times larger than the main peak in the map, the re-detection network will be initiated. The APCE and the GMM model achieve the goals to construct robust templates and update them adaptively

Secondly, the last layer has 32 rather than 128 output channels for maintaining the high processing speed.

3.1.2 SiamFC with re-detection mechanism

During the tracking process, SiamFC usually obtains a response map by comparing the similarity between the template patch and the search region. Ideally, the position with the largest response is the location of the object. However, due to the complex scenes and the similar interference, the object may appear at the location with non-maximum response value. To prevent this problem, the cosine window is used to suppress the edge

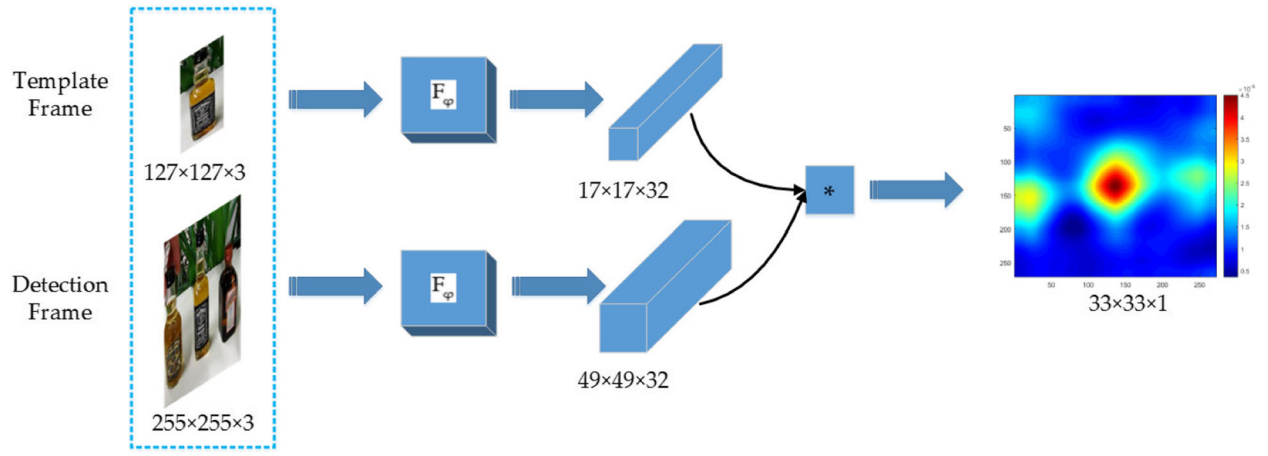

Fig. 3 The structure of SiamFC network. The network utilizes a template patch (T) and a larger search region (S) as inputs. The embedding function of the Siamese network is a feature extractorF_{φ}, which extracts the features of the template and the search region at the same time. Then, the network feeds the extracted features into the cross-correlation function and gets the final response map

Table 1 Architecture of the Baseline-conv5 model

Layer	Kernel Size	Stride	Action Size		Chans.
			For exemplar	For search	
			127 × 127	255 × 255	3
Conv1	11 × 11	2	59 × 59	123 × 123	96
Pool1	3 × 3	2	29 × 29	61 × 61	96
Conv2	5 × 5	1	25 × 25	57 × 57	256
Pool2	3 × 3	1	23 × 23	55 × 55	256
Conv3	3 × 3	1	21 × 21	53 × 53	384
Conv4	3 × 3	1	19 × 19	51 × 51	384
Conv5	3 × 3	1	17 × 17	49 × 49	32

response in SiamFC. The closer the peak value is to the center of the response map, the higher the weight is. Although this strategy greatly improves the stability of the algorithm, it also brings risks. When the object moves faster, the tracker with this strategy is easy to lose track of the object. Therefore, when more secondary peaks occur, a re-detection network is used to replace the cosine window.

The re-detection network mainly utilizes the SINT [36] framework, it trains the convolution network offline to get the matching function. According to the matching function, the candidate region which best matches the object in the initial frame is selected as the tracking result.

$$\hat{S}_T = \arg_{S_{j,T=t}} \max F\left(S_{T=0}, S_{j,T=t}\right) \tag{3}$$

where $S_{j,\ T=t}$ are candidate boxes in frame t, F is the matching function learned by the Siamese network, $S_{T=0}$ is the object in the first frame, $F(x, y) = F(x)^T F(y)$. By reducing the use of maximum pooling layer, the accuracy of SINT is much higher than that of SiamFC. At the same time, the framework employs a region-of-interest (ROI) pooling for fast processing multiple overlapping regions. This can not only greatly reduce the amount of computation, but also settle the problem of fixed patch size. The further understanding of the SINT tracker can be learned in [36].

The sketch map of the re-detection mechanism is shown in Fig. 4. Figure 4a is the response map of SiamFC with two distinct peaks. One of the peaks represents the response value of the real object, corresponding to the red box region in Fig. 4b. The other peak represents the response value of the interference, corresponding to the yellow box region in Fig. 4b. In such condition, the response intensity of the interference exceeds the intensity of the real object. In this paper, when

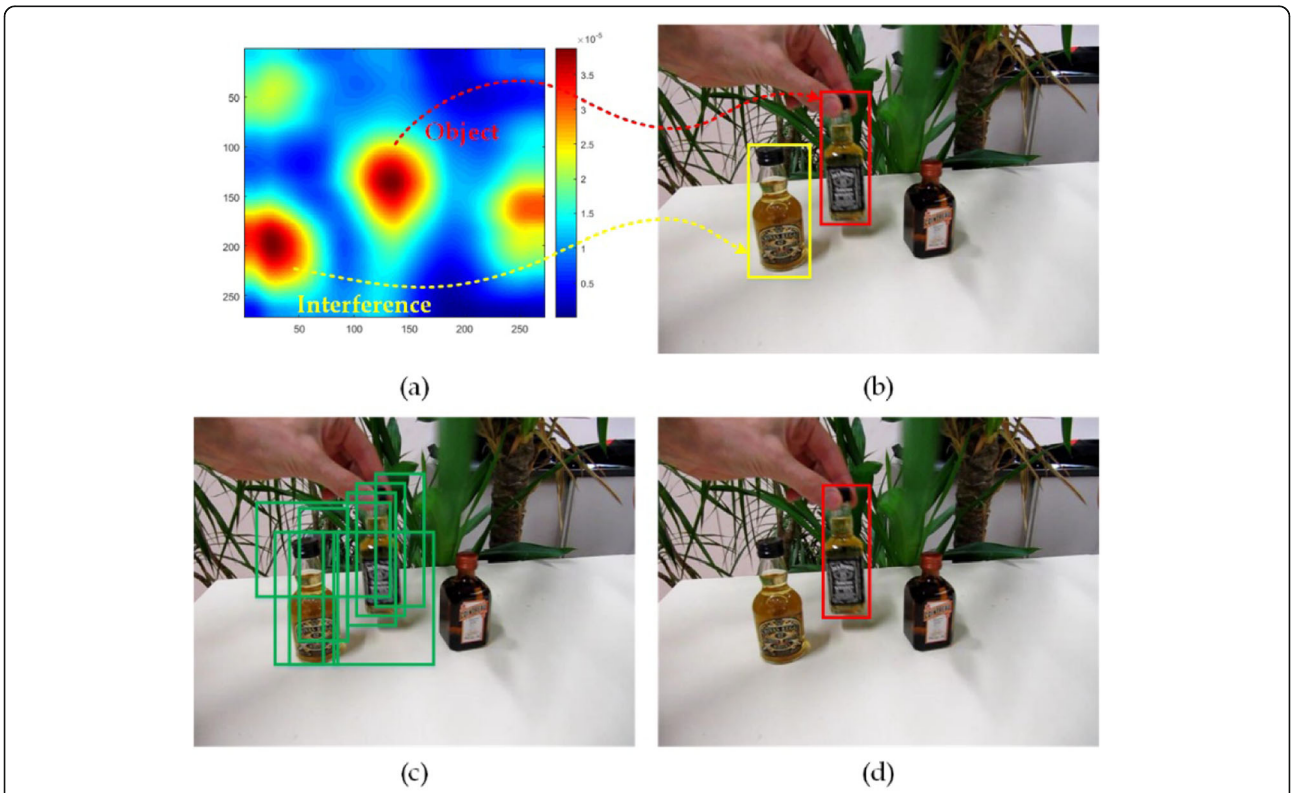

Fig. 4 The sketch map of the re-detection mechanism. **a** The response map of SiamFC with two distinct peaks. **b** The red box region is the real object and the yellow box region is the interference. **c** Candidates near the position of the peak value in the response map. **d** The tracking result

a secondary peak is 0.75 times larger than the main peak, the re-detection network will be initiated.

In order to make full use of the information of the response map, we randomly sample the candidates near the position of the main peak, as shown in Fig. 4c. Finally, these candidates will be detected by the SINT network to determine the exact location of the object. The result is shown in Fig. 4d.

3.2 Generative model for constructing templates

In most template-based trackers, templates always have limited modeling abilities since they only represent a single appearance of the object. When the appearance of the object changes greatly, the tracker will fail to track it. To solve this problem, a generative model based on Gaussian mixture model (GMM) is exploited for construction of the templates, instead of a unique and fixed template in the first frame. The probability distribution of template T is:

$$P(T) = \sum_{k=1}^{K} \pi_k N(T; \mu_k; I) \tag{4}$$

Here, K is the number of Gaussian components $N(T; \mu_k; I)$, π_k is the prior weight of the component k, μ_k is the mean of the component k. For avoiding costly inference in high-dimensional space, the covariance matrix in each component is set as an identity matrix I. The final template T_n will be constructed as:

$$T_n = (1-\eta) * T_g + \eta * \sum_{k=1}^{K} p(T_k) T_k \tag{5}$$

where η is the learning rate, T_k is the feature of the component k, T_g is the feature of the new component g, $p(T_k)$ is the probability of the new template corresponding to the existing component k. This probability can be calculated by the probability distribution of the GMM model. The illustration of the GMM model for constructing templates is shown in Fig. 5.

For effectively updating the GMM model, a simplified version of the online updating algorithm [53] is utilized. When a new template is available, we will first regard it as a new component g with $\pi_g = \eta$ and $\mu_g = e_g$. If the number of Gaussian components is less than K, the new template will be placed into the empty slot. If the number of components exceeds K, there are three possible cases to consider:

1. One of the components is outdated. In this case, we define the minimum weight of the template as $W_{\min}$. If the weight of the component is lower than $W_{\min}$, the template with the minimum weight will be replaced by a new one.
2. The minimum distance between the new component and the existing components is shorter than that between any two existing components. In this case, the new component will be merged with the nearest existing component.
3. The minimum distance between the new component and the existing components is longer than that between any two existing components. In this case, the two closest existing components are merged and the new component is placed in the free slot.

In above cases, the method of merging two closest components x and y can be modeled as:

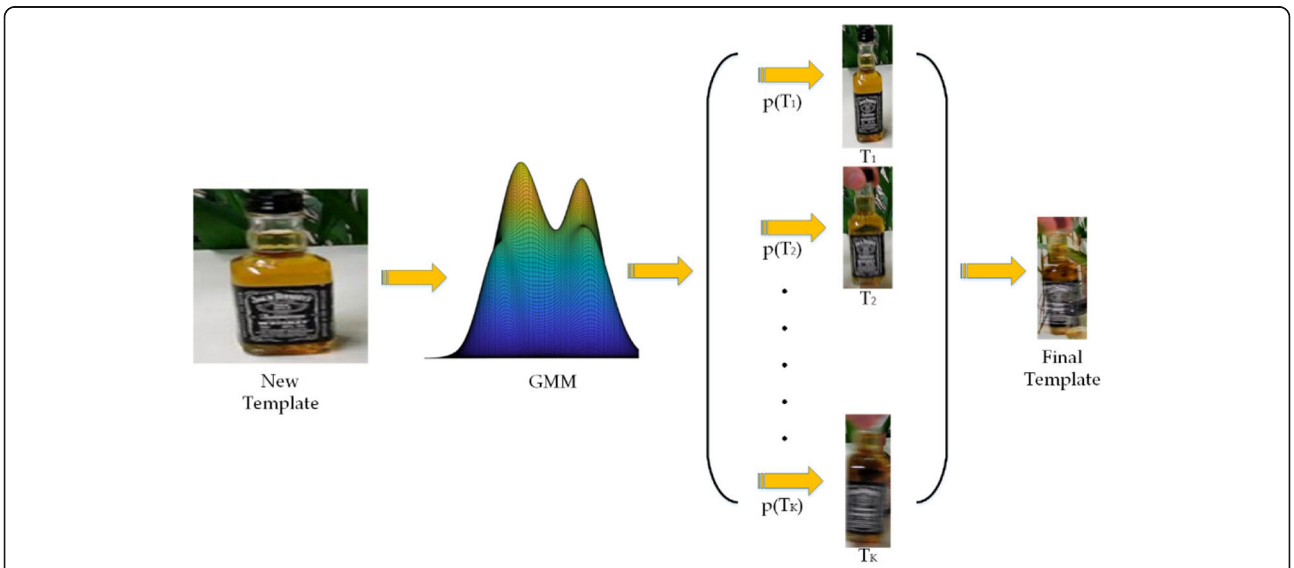

Fig. 5 The illustration of the GMM model for constructing templates. The GMM model utilizes the new template and the probability distribution corresponding to the existing component to construct templates

$$\pi_n = \pi_x + \pi_y \tag{6}$$

$$\mu_n = \frac{\pi_x \mu_x + \pi_y \mu_y}{\pi_x + \pi_y} \tag{7}$$

And the distance between two components is computed by using Euclidean distance.

3.3 Template updating

The standard approach in most template-based trackers is to update the template in each frame or utilize an evaluating mechanism [12, 25, 49, 50]. The former may have a severe impact on the computational load and the real-time performance will decline. The latter with low confidence will bring the noise of the template into the framework, which can severely affect the tracking accuracy.

Instead of updating the template in every frame or in a fixed interval, a strategy of template updating with high confidence is utilized to promote the quality of the template. In our framework, the level of the confidence depends on the quality of the response map. Inspired from [54], we adopt average peak-to-correlation energy (APCE) to measure the quality of the response map.

$$\mathrm{APCE} = \frac{|M_{\max} - M_{\min}|^2}{\mathrm{mean}\left(\sum_{w,h} \left(M_{w,h} - M_{\min}\right)^2\right)} \tag{8}$$

Here, $M_{\max}$ means the maximum value in the response map M, $M_{\min}$ means the minimum value in the response map M, and $M_{w,\ h}$ means the value of row w and column h in the response map M.

APCE can indicate the fluctuation degree and the confidence level of the response map. If there is a higher peak value and a smaller noise in the map, a larger APCE value will be obtained. Conversely, if there is an occlusion or similar interference, the APCE value will drop significantly. When the APCE value is larger than a certain threshold$\mathrm{Th}_{\mathrm{APCE}}$, we update the template by using the GMM model to prevent the template from being contaminated to a certain extent.

4 Experimental

4.1 Implementation details

4.1.1 Experimental environment

The proposed framework is implemented by using Matlab-R2016a with MatConvNet [55] and MatCaffe [56] toolbox. We perform the experiments on a PC with Intel i7-6850k CPU (3.60 GH), 64 GB RAM, and a single NVDIA GeForce GTX Titan X GPU. The average testing speed is 21fps.

4.1.2 Network structure

In SiamFC, we apply Baseline-conv5 model as the embedding functionF_ϕ. The architecture is shown in Table 1. Max-pooling layer is used to dispose the first two convolutional layers. Rectified linear unit (ReLU) follows each convolutional layer except the final layer conv5. The final strides of the network are 4 and the output channels in the final layer are 32. In re-detection network (SINT), we employ the AlexNet-like [51] network as the basic network. The detailed settings of the parameters in the network can refer to [36].

4.1.3 Training

The SiamFC and the SINT network are separately trained offline without online fine-tuning. In the process of training SiamFC, we use the ILSVRC-2015 [57] datasets to train the network. The initial values of the parameters in each layer follow a Gaussian distribution. Stochastic gradient descent method (SGD) with a weight decay of 0.0005 is used to optimize the network training. Both branches in SiamFC are trained for 50 epochs at a learning rate of 0.001. After each epoch, the learning rate is multiplied by a fixed factor until reaching 0.00001 in the final epoch. The momentum is 0.9 and the size of mini-batch is 32. For training SINT, we use the ALOV [58] dataset which covers many types of variations. We exclude 12 sequences that overlap with the online object tracking benchmark (OTB) [59]. To avoid training the Siamese network from scratch, we utilize the network pre-trained for ImageNet classification as the baseline. The learning rate of the network is 0.0001, the parameter of weight decay is 0.0005 and momentum is 0.9.

4.1.4 Other settings

For the generative model, which is presented in Section 3.2, we set the learning rate to$\eta = 0.011$. The number of components is set to$K = 30$ and the minimum weight of the template is set to $W_{\min} = 0.0036$. For the APCE which is used in Section 3.3, we set the certain threshold to $\mathrm{Th}_{APCE} = 8$.

4.2 Results and discussion

4.2.1 Quantitative comparison

For quantitatively evaluating the performance of our framework, we adopt the widely used OTB [59] sequences to implement the comparison. Two metrics are mainly employed for evaluation: center position error of the bounding box and overlap rate in the one-pass evaluation (OPE). Moreover, we use the precision plots and the success plots to show the results of the evaluation. The score in the precision plots is defined as the percentage of the frames in which the center position errors are smaller than the predefined threshold. The score in the success plots indicates the percentage of the frames in which the overlap rate of the tracking area and the boundary frame is larger than the threshold.

At the same time, we compare our tracker Siam-RM with nine state-of-the-art trackers, including SiamFC [35], SINT [36], CFNet [39], DCFNet [60], LCT [61] (long-term tracking algorithm with detection mechanism), LMCF [54], and real-time algorithms based on correlation filters (Staple [62], KCF [21], DSST [22]).

The results of the precision plots and the success plots are shown in Fig. 6. As can be seen from the figure, Siam-RM has great improvements compared with SiamFC. The tracking accuracy and the success rate of the proposed framework can reach 79.8% and 63.8%, respectively. Moreover, compared with SINT, the framework not only guarantees the real-time performance, but also improves the tracking precision.

The OTB sequences also present many challenging problems, including illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC), and low resolution (LR). For a detailed analysis, we also provide the precision plots (for clarity) of the attributes on the OTB sequences. Figure 7 demonstrates that Siam-RM ranks first in 8 attributes, especially in attribute LR which is far superior to other trackers.

In the attribute DEF, Siam-RM ranks second only after SINT. Though Siam-RM ranks third in attribute OV, it has a slight improvement over SiamFC. Compared with SiamFC, the proposed framework improves the precision of 11 attributes by 10.6% (BC), 7.7% (FM), 7.7% (IPR), 12.1% (IV), 16.3% (LR), 8.9% (MB), 5.9% (OCC), 7.4% (OPR), 4.9% (SV), 14.5% (DEF), and 0.2% (OV), respectively.

4.2.2 Qualitative comparison

To better analyze the performance of the tracking algorithm, we choose some challenging video sequences in Fig. 8 for further comparison. These video sequences represent different complex scenes. In order to show the results more clearly, six tracking algorithms are selected for comparison in this part. In each image, the boxes with different colors represent the corresponding tracking positions of different trackers. The legend shows the relationship between the different colors and the corresponding trackers.

4.2.2.1 FM For capturing the object when it moves fast, the small search area should not be used in the tracking algorithm. However, a large search area may increase the risk of introducing similar interference. The algorithms based on correlation filters have weak ability to distinguish the distant objects because the cosine windows are added in the training phase.

In Deer and Liquor sequence, the moving speed of the target is fast. The target blends with the background and changes abruptly. Some trackers have positioning errors (LCT, KCF) or stay in the previous position (KCF, LCT, CFNet), but Siam-RM can track stably throughout the entire process.

In Jumping sequence, the object jumps up from frame 33. In frame 39, LCT, Staple, and KCF lose the object first. Staple relocates the object in frame 41, but fails to track in frame 146. Because of the addition of the cosine window into the response map, SiamFC suppresses the response of the object at the edge of the search area. However, two frames are often required in subsequent frames to keep up with the fast-moving object. Siam-RM removes the cosine window and a re-detection network is used to detect the multiple peaks. This not only enables the tracker to resist similar interference even in a large search area, but also ensures the tracking accuracy of the framework.

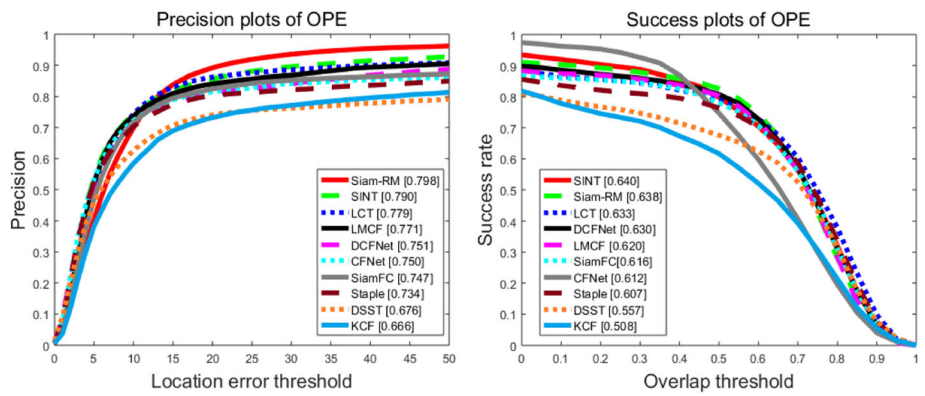

Fig. 6 The precision plots and the success plots of OPE for top 10 trackers. Each tracker is ranked by the score of tracking performance. For precision plots, the results at the error threshold of 20 are used for ranking. For success plots, the area under the curve (AUC) is used to rank the trackers

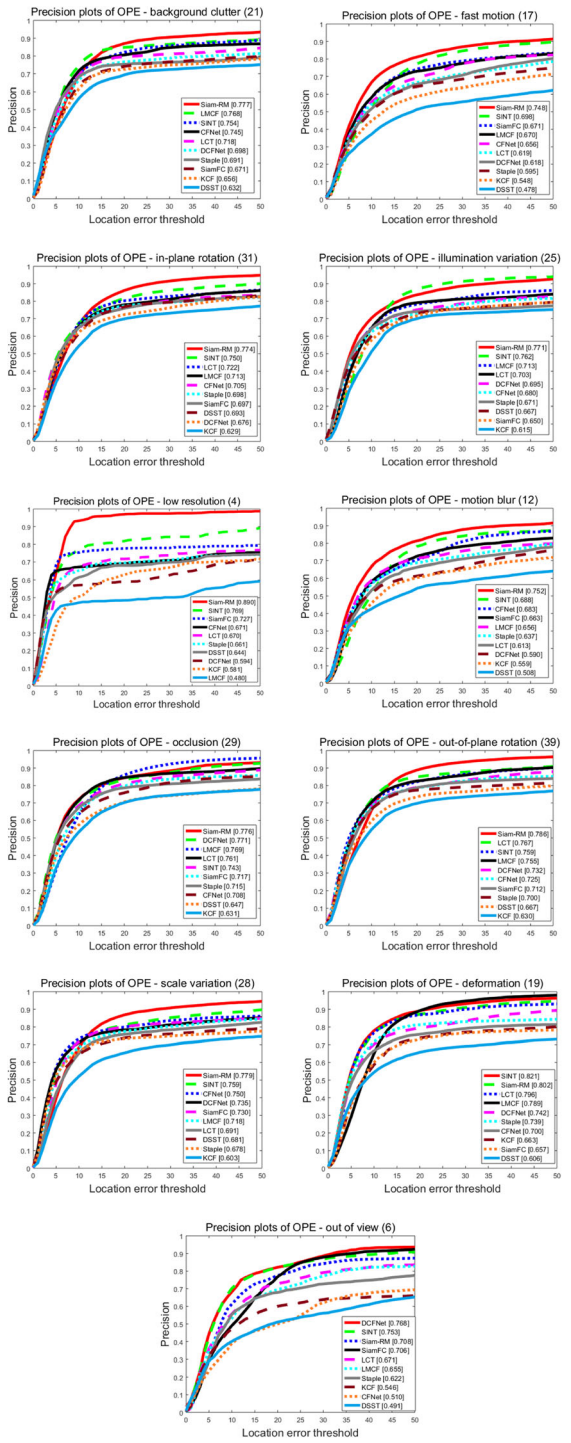

Fig. 7 Precision plots of OPE in 11 attributes. The OTB sequences can be divided into 11 attributes, including illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC), and low resolution (LR). The precision plots of each attribute represent the performance in the current attribute

4.2.2.2 OCC Occlusion is a major challenge in target tracking. If the object is occluded without a valid template strategy, the tracker will mistake the occlusion as the object.

In Freeman4 sequence, the target is almost completely occluded from frame 157. When the target reappears in frame 161, Siam-RM and SiamFC first locate the target. SiamFC does not update the template and loses the target in frame 202. Siam-RM utilizes a strategy of template updating strategy with high confidence to avoid the templates being contaminated. KCF and Staple confirm the occlusion region as the target and lose the real target.

In Soccer sequence, starting from frame 95, the target has been severely occluded. All the trackers are in a random state. In frame 120, the target reappears in the field of view and CFNet first locates the object. In frame 144, Siam-RM also locates the target by adopting the template with high confidence and the re-detection mechanism. Other trackers fail to track until the end.

In Football sequence, the target is severely occluded from frame 115. In frame 310, KCF and Staple track the similar interference next to the target. By utilizing the GMM model and a high-confidence updating strategy to construct the template, Siam-RM achieves a more accurate performance than other trackers.

4.2.2.3 BC and IV Background clutter means that the color or the texture of the background near the target is similar to that of the target, which requires a tracker with high discriminating ability. By adopting CNN features, Siam-RM, SiamFC, and CFNet perform better than other trackers in discriminating similarity. The illumination variation is easier to overcome than the background clutter in some uncomplicated scenes, but when combined with the background clutter, the tracking results will easily drift to the similar interference near the target.

In Ironman, Matrix, Shaking, and CarDark sequences, IV and BC are both included. As can be seen from the results in Fig. 7, trackers with correlation filters (LCT, Staple, and KCF) cannot track accurately. They often fail to track when the illumination changes greatly or the background clutter is large. By contrast, the trackers with CNN features (Siam-RM, SiamFC, CFNet) have better tracking performance. These trackers can not only effectively distinguish the target from the background, but also prevent the influence of illumination variation on feature vectors. However, due to the lack of effective re-detection mechanism and reliable template strategy, the bounding boxes of SiamFC and CFNet often drift to the background. Siam-RM utilizes the GMM model to solve the problem of changes in appearance which is caused by illumination variations. This model uses all reliable

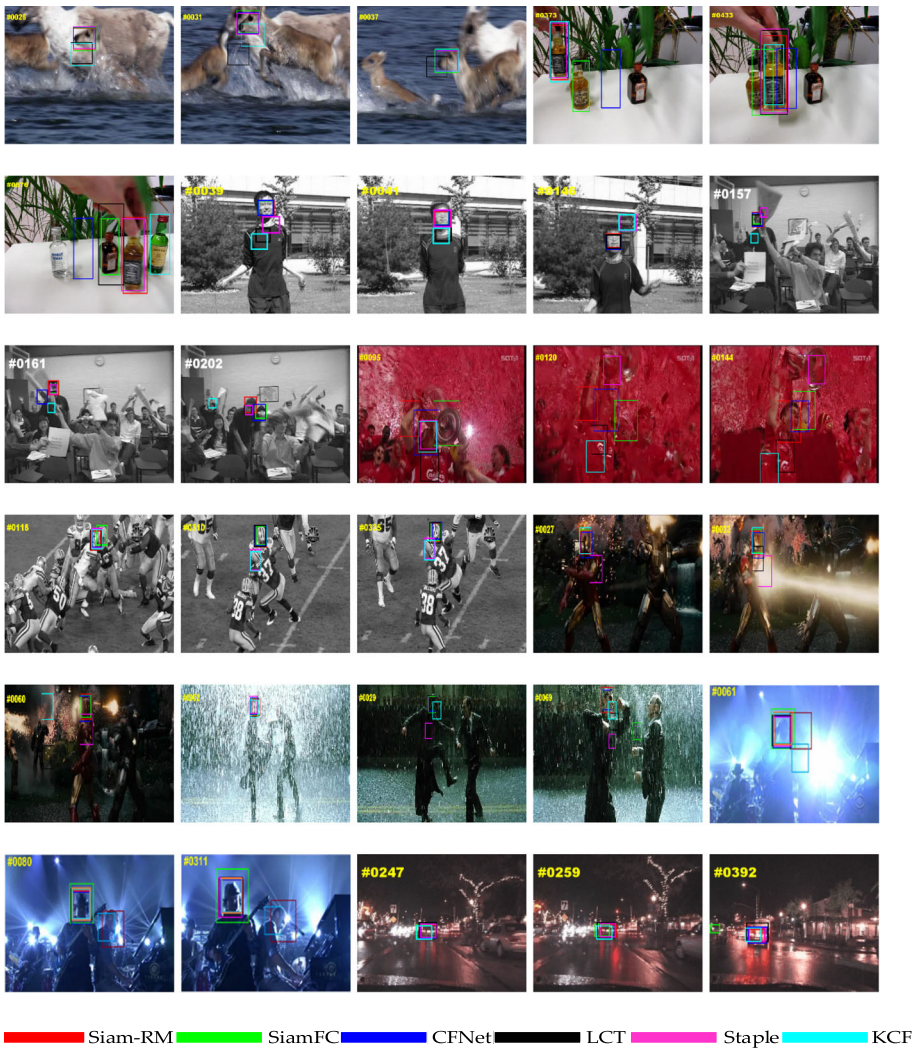

Fig. 8 Tracking results of the sequences. For a further clear comparison, some challenging video sequences are selected from all OTB sequences. These video sequences represent different complex scenes. In order to show the results more clearly, six tracking algorithms are selected for comparison in this part. In each image, the boxes with different colors represent the corresponding tracking positions of different trackers. The legend shows the relationship between the different colors and the corresponding trackers

templates to construct a new information-rich template. A strategy of template updating with high confidence is also adopted to avoid the templates being contaminated. Moreover, a more precise re-detection mechanism can effectively prevent the interference of background clutter. Thus, the tracking performance can remain stable and accurate.

4.2.3 Computing latency

The proposed tracker (Siam-RM) can be regarded as an iterative mechanism and the average testing speed is 21 fps. Unlike other conventional CNN-based trackers, Siam-RM removes the fully connected layer to reduce the amount of computation. Moreover, only five fully convolutional layers are applied to train an end-to-end Siamese network for learning a similarity function. The function is directly used for online tracking without any complex fine-tuning tactics. Thus, the computing latency of Siam-RM can be effectively decreased. However, compared with the computing latency of SiamFC which is the baseline of the proposed tracker, the computing latency of Siam-RM increases significantly due to the introduction of the feedback loop, and the processing speed decreases from 65 fps to 21 fps. In Siam-RM, the re-detection mechanism will be activated when the secondary peak is 0.75 times larger than the main peak. Meanwhile, the GMM model and APCE strategy achieve the goals to construct robust templates and update them adaptively. Although these proposed methods improve the tracking performance of Siam-RM based on SiamFC, the computing latency

increases as well. Therefore, in future studies, we will further reduce the latency while ensuring the tracking performance of the tracker.

5 Conclusion

In this paper, we aim at improving the deficient tracking ability of SiamFC in complex scenes with fast motion and similar interference. When multiple peaks appear on the response map of SiamFC, a more accurate re-detection network can re-determine the location of the object. Meanwhile, for adapting to various changes in appearance of the object, we employ the GMM model to construct the template. Furthermore, a strategy of template updating with high confidence is also used to prevent the template being contaminated. The objective evaluation on the OTB sequences shows that the tracking accuracy and the success rate of the proposed framework can reach 79.8% and 63.8%, respectively. Compared to SiamFC and other state-of-the-art trackers, the results of several representative video sequences demonstrate that our framework has higher accuracy and robustness in scenes with fast motion, occlusion, background clutter, and illumination variations. The next step of our work is to improve the re-detection network and the real-time performance of the framework, so as to achieve the application of CNNs-based trackers in practical engineering.

Abbreviations

APCE: Average peak-to-correlation energy; BC: Background cluttered; CNNs: Convolutional neural networks; DEF: Deformation; FM: Fast motion; GMM: Gaussian mixture model; IPR: In-plane rotation; IV: Illumination variation; LR: Low resolution; MB: Motion blur; NCC: Normalized cross-correlation; OCC: Occlusion; OPR: Out-of-plane rotation; OTB: Online tracking benchmark; OV: Out-of-view; SiamFC: Fully-convolutional Siamese networks; SINT: Siamese instance search tracker; SV: Scale variation

Acknowledgements

The authors would like to thank Yi Yu for the support.

Authors' contributions

DL and YY designed the global structure and the experiments. DL performed both software experiments. YY and XC analyzed the data. DL wrote the paper. All authors read and approved the final manuscript.

Authors' information

Not applicable

Funding

This work was supported by the General Program of National Nature Science Foundation of China under grant 51675506.

Availability of data and materials

All data are fully available without restriction.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Chinese Academy of Sciences, Changchun Institute of Optics Fine Mechanics and Physics, 3888 Dongnanhu Road, Changchun 130033, China. [2]University of Chinese Academy of Sciences, 19 Yuquan Road, Beijing 100049, China.

Received: 15 May 2019 Accepted: 16 October 2019
Published online: 29 November 2019

References

1. H. Fan, H. Ling, *SANet: Structure-aware network for visual tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
2. J. Choi, H.J. Chang, T. Fischer, et al., *Context-aware deep feature compression for high-speed visual tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018)
3. T. Yang, A.B. Chan, *Recurrent filter learning for visual tracking* (IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017)
4. A. Toshev, C. Szegedy, *DeepPose: Human pose estimation via deep neural networks* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, 2014)
5. Y. Ioannou, D. Robertson, R. Cipolla, et al., *Deep roots: improving CNN efficiency with hierarchical filter groups* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
6. V. Lepetit, P. Fua, Keypoint recognition using randomized trees. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1465–1479 (2006)
7. M. Ozuysal, P. Fua, V. Lepetit, *Fast keypoint recognition in ten lines of code* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, 2007)
8. M. Calonder, V. Lepetit, P. Fua, *BRIEF: Binary Robust Independent Elementary Features* (European Conference on Computer Vision (ECCV), Heraklion, 2010)
9. Z. Kalal, K. Mikolajczyk, J. Mata, *Forward-backward error: automatic detection of tracking failures* (International Conference on Pattern Recognition, Istanbul, 2010)
10. S. Hare, A. Saffari, P.H. Torr, *Struck: structured output tracking with kernels* (IEEE International Conference on Computer Vision (ICCV), Barcelona, 2011)
11. Y. Zhang, B. Ghanem, S. Liu, et al., *Robust visual tracking via multi-task sparse learning* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, 2012)
12. D.A. Ross, J. Lim, R.S. Lin, et al., Incremental learning for robust visual tracking. International Journal of Computer Vision **77**(1-3), 125–141 (2008)
13. T.W. Wang, I.Y.H. Gu, P. Shi, *Object tracking using incremental 2D-PCA learning and ML estimation* (IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, 2007)
14. X. Mei, H. Ling, *Robust visual tracking using l1 minimization* (IEEE International Conference on Computer Vision (ICCV), Kyoto, 2009)
15. B. Han, D. Comaniciu, Y. Zhu, et al., Sequential kernel density approximation and its application to real-time visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. 30(7), 1186–1197 (2008)
16. B. Babenko, M.H. Yang, S. Belongie, Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 33, 1619–1632 (2010)
17. K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning. Pattern Recognition 46(1), 397–411 (2013)
18. H. Grabner, M. Grabner, H. Bischof, *Real-time tracking via on-line boosting* (Proc of British Machine Vision Conference (BMVC), Edinburgh, 2006)
19. S. Avidan, Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007)
20. D.S. Bolme, J.R. Beveridge, et al., *Visual object tracking using adaptive correlation filters* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, 2010)
21. J.F. Henriques, R. Caseiro, P. Martins, et al., High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596 (2015)
22. M. Danelljan, G. Hager, F. Khan, *Accurate scale estimation for robust visual tracking* (Proc of British Machine Vision Conference (BMVC), Nottingham, 2014)
23. Z. Hong, Z. Chen, C. Wang, et al., *MUlti-store tracker (MUSTer): a cognitive psychology inspired approach to object tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015)
24. M. Danelljan, G. Hager, F.S. Khan, et al., *Learning spatially regularized correlation filters for visual tracking* (IEEE International Conference on Computer Vision (ICCV), Santiago, 2015)
25. Z. Kalal, K. Mikolajczyk, J. Mata, Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 34, 1409–1422 (2012)
26. M. Danelljan, G. Bhat, F.S. Khan, et al., *ECO: Efficient convolution operators for tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)

27. M. Danelljan, G. Hager, F.S. Khan, et al., *Convolutional features for correlation filter based visual tracking* (IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, 2015)
28. M. Danelljan, A. Robinson, F.S. Khan, et al., *Beyond correlation filters: learning continuous convolution operators for visual tracking* (European Conference on Computer Vision (ECCV), Amsterdam, 2016)
29. Q. Yuankai, Z. Shengping, Q. Lei, et al., *Hedged deep tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)
30. B. Han, J. Sim, H. Adam, *BranchOut: Regularization for online ensemble tracking with convolutional neural networks* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
31. H. Nam, M. Baek, B. Han, *Modeling and propagating CNNs in a tree structure for visual tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
32. H. Nam, B. Han, *Learning multi-domain convolutional neural networks for visual tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)
33. L. Wang, W. Ouyang, X. Wang, et al., *STCT: Sequentially training convolutional networks for visual tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)
34. D. Held, S. Thrun, S. Savarese, *Learning to track at 100 FPS with deep regression networks* (European Conference on Computer Vision (ECCV), Amsterdam, 2016)
35. L. Bertinetto, J. Valmadre, J.F. Henriques, et al., *Fully-convolutional siamese networks for object tracking* (European Conference on Computer Vision (ECCV), Amsterdam, 2016)
36. R. Tao, E. Gavves, A.W.M. Smeulders, *Siamese instance search for tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)
37. C. Huang, S. Lucey, D. Ramanan, *Learning policies for adaptive tracking with deep feature cascades* (IEEE International Conference on Computer Vision (ICCV), Venice, 2017)
38. Q. Guo, W. Feng, C. Zhou, et al., *Learning dynamic Siamese network for visual object tracking* (IEEE International Conference on Computer Vision (ICCV), Venice, 2017)
39. H. Xu, Y. Gao, F. Yu, et al., *End-to-end learning of driving models from large-scale video datasets* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
40. X. Wang, Greedy batch-based minimum-cost flows for tracking multiple objects. IEEE Transactions on Image Processing. **26**(10), 4765–4776 (2017)
41. A. Maksai, X. Wang, F. Fleuret, et al.: Globally consistent multi-people tracking using motion patterns. arXiv:1612.00604, 2016 [Online]. Available: https://arxiv.org/abs/1612.00604
42. A. Maksai, X. Wang, P. Fua, *What players do with the ball: a physically constrained interaction modeling* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)
43. E. Kodirov, T. Xiang, Z. Fu, *Person re-identification by unsupervised l1 graph learning* (European Conference on Computer Vision (ECCV), Amsterdam, 2016)
44. W.S. Zheng, S. Gong, Xiang, T.:Towards open-world person re-identification by one-shot group-based verification. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 591–606 (2016)
45. J.S. Supancic, D. Ramanan, *Self-paced learning for long-term tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, 2013)
46. Y. Hua, K. Alahari, C. Schmid, *Occlusion and motion reasoning for long-term tracking* (European Conference on Computer Vision (ECCV), Zurich, 2014)
47. N. Wang, W. Zhou, H. Li, Reliable re-detection for long-term tracking. IEEE Transactions on Circuits and Systems for Video Technology **29**(3), 730–743 (2019)
48. S. Oron, A. Bar-Hille, S. Avidan, *Extended Lucas-Kanade tracking* (European Conference on Computer Vision (ECCV), Zurich, 2014)
49. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell., 2003, 25(10), 0-1311.
50. Matthews, L., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. Pattern Anal. Mach. Intell. 2004, 26(6):0-815.
51. A. Krizhevsky, I. Sutskever, G.E. Hinton, in *Advances in neural information processing systems*. Imagenet classification with deep convolutional neural networks (2012), pp. 1097–1105
52. J. Valmadre, L. Bertinetto, J.F. Henriques, et al., *End-to-end representation learning for correlation filter based tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
53. A. Declercq, J.H. Piater, *online learning of gaussian mixture models-a two-level approach. VISAPP* (Proceedings of the Third International Conference on Computer Vision Theory and Applications, Funchal, 2008)
54. M. Wang, Y. Liu, Z. Huang, *Large margin object tracking with circulant feature maps* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017)
55. A. Vedaldi, K. Lenc, *MatConvNet - convolutional neural networks for MATLAB* (23rd ACM International Conference on Multimedia, Brisbane, 2015)
56. Y.Q. JIA, E. SHELHAMER, J. DONAHUE, et al., *Caffe: convolutional architecture for fast feature embedding* (22nd ACM International Conference on Multimedia, Orlando, 2014)
57. O. Russakovsky, J. Deng, H. Su, et al., ImageNet large scale visual recognition challenge. International Journal of Computer Vision. **115**(3), 211–252 (2014)
58. A.W. Smeulders, D.M. Chu, R. Cucchiara, et al., S.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. 36(7), 1442–1468 (2014)
59. Y. Wu, J. Lim, M.H. Yang, *Online object tracking: a benchmark* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, 2013)
60. Q. Wang, J. Gao, J. Xing, et al., DCFNet: discriminant correlation filters network for visual tracking. arXiv preprint arXiv 1704 04057 (2017)
61. C. Ma, X. Yang, N.C. Zhang, et al., *Long-term correlation tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, 2015)
62. L. Bertinetto, J. Valmadre, S. Golodetz, et al., *Staple: Complementary learners for real-time tracking* (IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016)

Publisher's Note