electronics

Review

Model Update Strategies about Object Tracking: A State of the Art Review

Deyu Wang [1,2], Weidong Fang [3], Wei Chen [1,2,4,*], Tongfeng Sun [1,2], and Tingjie Chen [1,2]

[1] School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221000, China; hnsdwdy@126.com (D.W.); suntf@cumt.edu.cn (T.S.); 2390@cumt.edu.cn (T.C.)

[2] Mine Digitization Engineering Research Center of the Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China

[3] Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Micro-System and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China; weidong.fang@mail.sim.ac.cn

[4] School of Earth and Space Sciences, Peking University, Beijing 100871, China

[*] Correspondence: chenwdavior@163.com

Received: 24 September 2019; Accepted: 17 October 2019; Published: 23 October 2019

Abstract: Object tracking has always been an interesting and essential research topic in the domain of computer vision, of which the model update mechanism is an essential work, therefore the robustness of it has become a crucial factor influencing the quality of tracking of a sequence. This review analyses on recent tracking model update strategies, where target model update occasion is first discussed, then we give a detailed discussion on update strategies of the target model based on the mainstream tracking frameworks, and the background update frameworks are discussed afterwards. The experimental performances of the trackers in recent researches acting on specific sequences are listed in this review, where the superiority and some failure cases on each of them are discussed, and conclusions based on those performances are then drawn. It is a crucial point that design of a proper background model as well as its update strategy ought to be put into consideration. A cascade update of the template corresponding to each deep network layer based on the contributions of them to the target recognition can also help with more accurate target location, where target saliency information can be utilized as a tool for state estimation.

Keywords: visual tracking; update occasion; update mechanism; background model; network layer contribution; saliency information

1. Introduction

With the progress of computer vision technology, moving target tracking is being increasingly popularly researched, which has become a challenging topic in the area of smart application. As the development of computer hardware devices and rapid progress of machine learning and deep learning techniques, researches on each respect of moving target tracking has been endowed with great essence. Object tracking has been greatly related to many applications in modern life, i.e., player identification, vehicle monitor, smart human-computer interactions [1]. The mechanism of tracking a moving target is that the target, which is distinguishable from the background, is separated out and marked by a bounding box, which is usually regarded as a classification issue that target samples and background ones should be from different classes. Nowadays, lots of frameworks of image classifiers, i.e., support vector machine (SVM) [2], extreme learning machine (ELM) [3], Integrated Circulant Structure Kernels (ICSK) [4], etc., have been widely utilized for researches of visual tracking. Furthermore, deep learning is getting more and more popularly concerned, trackers using which framework have gained more excellent performances due to the development of neuroscience.

Diverse variations regarding the target usually occur in the process of tracking, i.e., variations arise from changes of the outside environment, such as view angle, camera orientation, environmental illumination, etc., and inherent changes of the object, such as self-rotation, self-deformation, and self-variation of target appearance; therefore, a tracker with more robust capacity has to be designed, whose framework structure and sample learning strategy are of key importance, which guarantees its real-time and accuracy. Consequently, researching an update strategy with higher robustness and efficiency has been of greater essence.

Object tracking framework can be usually typed into two categories: generative frameworks and discriminative ones, where, for the former framework, i.e., particle filter, sparse coding, linear predictions [5,6], Kalman filter, etc., target and background models are established at the beginning and the features of them are extracted for the search of similar target or background features in succeeding frame images to iteratively locate the target; The latter, i.e., deep neural networks, correlation filter, random forest, feature bagging [7], etc., gets the object location by drawing candidate target patches within a region and then select one that is distinguished from given background patches. With the progress of researches on machine learning and deep learning tracking frameworks, the model update has become a widely concerned part in recent researches. A good update mechanism is a crucial respect measuring the reliability of a tracker. On the one hand, template models of the target and background should be constantly updated to catch up with the their variation, which is a fundamental requirement of model adaptation. On the other hand, the parameter model must be adjusted with the same pace of the variations of the samples to satisfy the real-time requirement. Generally speaking, when and how to update make the major parts of the update task. In general, a less-frequent update cannot make sure that the target model can catch up with the change of target appearance, which gives rise to tracking failure, while much too frequent update makes it excessively adaptive to new characteristics of targets but neglects the influences of historical ones, which leads to background drift after a sudden occlusion comes across, thus incurring fatal errors. Up to now, specific update methods are designed to deal with tracking under irregular situations, such as occlusion and background clutter. For instance, more attention will be given to the background analysis when partial occlusion occurs. Although model update technology of visual tracking is gaining rapid progress and has obtained substantial break through at present, there are too few reviews about it compared to other works of tracking, as most reviews still focus on model construction and mathematical algorithms. This review will provide discussions on recently-proposed model update mechanisms and talk about the merits and drawbacks of them. Measures of improvement based on the superiority of existed update strategies and the remaining challenging tracking problems are proposed at the end of this paper. The remaining part of this paper is organized, as follows:

In Section 2, target update occasions in recent researches are talked about, in which three common tools—occlusion detection, response map, and similarity judgement—and two complementary update occasions—conservative update and long-short-term update (LST)—are respectively discussed in detail. In Section 3, the update strategies of target models are illustrated, where recent strategies under four commonly used frameworks—correlation filter (CF), dictionary sparse coding, bag-of-words (BoW), and deep neural network—are respectively analyzed in detail. Background update mechanisms are then illustrated in Section 4, where a new background update framework, called tracking with background estimation (TBE), is briefed. In Section 5, tracking experiment performances of recent trackers are listed, afterwards superior performances under several challenge factors of each typical tracker and some failure cases are exampled and analyzed. Specific conclusions regarding the update mechanisms are drawn from the testing statistics, and improvement measures of model update are briefly summarized in Section 6.

2. Review on Target Model Update Occasions

Determination of the model update occasion is a key part of the update process. Low-frequency update makes it difficult for a tracker to adapt to variations of target appearance, while too frequent update might make the target model introduce too much newest bounding box information that increases the probability of background drift, meanwhile datum calculation burden grows, which cuts down the tracker's efficiency. In general, update occasions often embodies the types below:

(1) update frame-by-frame;
(2) update for every certain amount of frames;
(3) update when the target response is higher than a threshold; and,
(4) update when the target becomes less distinguished from the background.

Generally, the method that to merely update for every certain period neglects the distinction of the target variation and its response, as well as the consideration of dealing with wrong updates, which makes the tracker update too frequently when the target appearance remains stable for quite a long period or update less frequently if the target constantly changes it appearance, which gives rise to error accumulation that leads to tracking drift. Therefore, trackers with this kind of update method have less robustness. Though update frame-by-frame, i.e., correlation filter, might well make the model tightly pace with the variation of the target, this kind of update unavoidably brings about calculation burden, thus lengthening the datum processing time, incurring unnecessary troubles to some extent. Accordingly, to speed up the calculation, Fast Fourier Transmission (FFT) and Kernelized Correlation Filter (KCF) have been recently proposed that are usually combined with the traditional correlation filter method for image procession. For the construction of a more robust tracker that can pace with target appearance variation as well as avoid error accumulation that is caused by improper update and decrease calculation burden, mere frame-by-frame update or updating with a fixed time interval is rarely adopted in recent researches, hence lots of target response assessment mechanisms, i.e., response maps, foreground and background histogram, multiple-class dictionaries, etc., are proposed. Once the tracked target in a frame is regarded as responsible, target the model update is then enabled, otherwise the tracked object has less responsibility and model update is temporarily stopped.

2.1. Update Using Occlusion Detection

Occlusion is one of the most challenging factors in the process of tracking. It is unavoidable that information of the occluding background part will be integrated into the target model if mere frame-by-frame or fixed-time-interval update is adopted, which makes the tracker mistakenly detect the occluding background part as the target, thus the bounding box stops at the occluding part [8]. Therefore, occlusion detection is required for judging whether the target has been occluded. Occlusion comprises of partial occlusion and full occlusion. In the latter case, almost all of the pixels in the view are background, which means that the target has temporarily disappeared. It is hardly possible to observe the variation of the target's appearance, so target model update is usually stopped when full occlusion happens. However, when the target is partial occluded, only a part of it is visible, hence part of the pseudo target information can be mixed with the target one in the target model if the regular update mode is still used in this case. A special update mode should be utilized in the case of partial occlusion.

There are increasing researches dealing with occlusions in recent year. Although it is easy for the tracker to identify whether the target is under full occlusion, partial occlusion or no occlusion, in quite a few researches, the update is only enabled when there is no occlusion, while it is disabled if partial occlusion happens. For instance, several small patches will be drawn within and around the bounding box after the target is located in a frame in [9] and the patches are classified into three types, where the patches from class #A do not overlap with the bounding box at all, while those from class #B overlap with the bounding box with higher target response and class #C with lower target response. The target is regarded to be occluded if the number of patches from class #C reaches the threshold, thus the target model is prohibited. Conventional correlation filter model update method is adopted in [10], where the fixed learning rate is used for target appearance model update when there is no occlusion; otherwise, the appearance model remains unmodified. Similar strategy is utilized in [11] for target occlusion detection, in which the occluding coefficient of each patch is calculated after the target is located. An update is disabled when the sum of the coefficients is above a given threshold. Complementary features, histogram of oriented gradient (HoG) and Hue, Saturation, Value (HSV), are used in [12] for tracking, where templates that are related to HoG and HSV are respectively established. Background pixel masking is carried out when there is occlusion and target's accurate position and scale is further calculated when partial occlusion happens. Still, the update of two feature templates is enabled only if the target undergoes no occlusion. The Bhattacharyya Distance between the candidate filters and the template in [13] has been used to identify occlusion in this research. Occlusion happens if the distance is above a threshold and thus the template is no updated.

Although the conservative update strategy that target model update is prohibited when the target is partial occluded can well prevent background patches from contaminating target templates, the probability of target appearance variation in each frame never equals to zero, even if the target is in the status of occlusion, therefore if the appearance model of the target is not properly updated at this stage, the tracker might also be unable to pace with the change of the target, thus losing the tracking before the target completely disappears. Local patterns are commonly used in some works to solve the problem of target model update under partial occlusion. In the framework of local patterns, a target model is departed into multiple non-overlapped patches, each of which is respectively tracked to alleviate the impact of pseudo targets. In order to use local information of a target while remaining the holistic structure under the situation of partial occlusion, local tracking that integrates holistic patch and local ones is utilized in [14], in which a tracked object is departed into seven patches, including a global one. The contribution score of each patch is calculated after it is tracked in a frame; afterwards, patches with larger score will be selected for model retraining. To make use of available features of unoccluded parts, in [15], part-based tracking that is similar to the idea in [14] is employed in the state of partial occlusion. Key feature points are extracted to construct the target Gaussian map to obtain the number of patches, thus the correlation filter of each patch is defined. Note that mere global pattern is still utilized when the target is not occluded. For the recovery of a target after full occlusion, owing to the fact that important target information has been preserved by the ICSK model in [4] at the moment before the period of full occlusion, it is usually essential to use the information of the target in the frames before full occlusion, after all of this period belongs to partial occlusion. To preserve the important target information, detected object samples are still selected to update the classifier when the target is partial occluded thanks to the ability to determine scale and position of ICSK, meanwhile ICSK parameters are also preserved. During full occlusion, the parameter set of the optimal classifier is selected according to the energy formulation to identify the reappearance of the target.

The tracked target cannot be identified as being completely responsible, as background pixels may exist together with foreground ones in the bounding box more or less. Even though the background pixel masking process [15] can help to alleviate the interference of background pixels, the existence of noise might not ensure the correct mask of each pixel, thus the background-removed foreground template might not be credible. Up to now, many frameworks, such as dictionary learning (DL) and sparse coding (SC), utilize multiple-class and local-representation structures, i.e., local background and foreground dictionaries are respectively modeled to check out how much background information takes up in the representation of a tracked target so as to correctly track unoccluded parts of a target and enhance the ability to discriminate the background from foreground of some generative models. Owing to the sparsity of image information during partial occlusion, visible parts of the tracking result are used for the encoding of template patches [16], where the corresponding template patches less represent the occluded parts and other parts are regularly updated. Three types of dictionaries are constructed in [17], namely $\mathbf{D}$, $\mathbf{D}_o$, and $\mathbf{D}_b$, which respectively donate the tracking dictionary, target dictionary, and background dictionary to enhance the ability to separate the background from foreground for better target locations. A tracking result is classified into three types of patches, namely stable patches, valid patches, and invalid patches, in which a stable patch is constantly represented by the patch at the same region of the template during some period, while valid ones are the patches that are represented with less error by foreground template patches than background ones and invalid ones are more frequently represented by background template patches. A tracking result is regarded as reliable when the number of valid patches is no less than an extent and the total number of valid patches and stable ones is also no less than a certain threshold; therefore, **D** and $\mathbf{D}_o$ are respectively updated, in which $\mathbf{D}_o$ is updated while using valid patches.

2.2. Update Using Response Maps

To judge the responsibility of a tracked object, in the past two years, response maps have been widely utilized in the field of visual tracking. A response map shows the probability of each pixel belonging to the target, whose maximum value point is near to the center of the Gaussian map of the target when the target is normally tracked, and when it is projected to a three-dimensional coordinate, it appears to have only one sharp peak around which the values sharply decrease with farther distance to it. When occlusion or background clutter comes across, more than one peak value can appear in the same response map, or even there is only one peak, the peak appears not so high enough or it is not sharp enough. Processed forms of the response map i.e., *PSR*, *PAR*, *APCE*, etc., are widely adopted in some researches to identify the presence of occlusion or background clutter, which are the derived parameters that measure the responsibility of a tracking result.

A tracking result is only judged to be reliable when the three-dimensional (3-D) response map of the frame image has only one sharp peak. *PAR* [18] is defined to represent the fluctuation of a response map to reflect the reliability of a tracked target, whose formulation is

$$PAR = \frac{R_{\max}^2}{\mathrm{mean}(\sum_{w,h} R_{w,h}^2)}, \tag{1}$$

in which $R_{\max}$ represents the maximum response value, $R_{w,h}$ is the value at a specific position, and the mean function calculates the average value of the map. Higher *PAR* indicates a more reliable tracking result. When the *PAR* and $R_{\max}$ are both greater than a predefined threshold, the result is judged as reliable, thus the correlation filter model in [18] is updated. Similarly, *APCE* is defined in [19], as

$$APCE = \frac{(F_{\max} - F_{\min})^2}{\mathrm{mean}(\sum_{w,h} (F_{w,h} - F_{\min})^2)}, \tag{2}$$

where $F_{\max}$ and $F_{\min}$, respectively, denote the maximum and minimum value of the response map, and this parameter also reflects the fluctuation of the map. The context correlation filter in [12] is updated when *APCE* and $F_{\max}$ are both higher than the threshold.

The parameter *PSR* is also similarly defined, except that the sharpness of the peak is not put into consideration, which is calculated by firstly subtracting the mean value and then dividing by the standard deviation, as (3) in [20]

$$PSR = \frac{R_{\max} - \mu}{\sigma}, \tag{3}$$

where μ and σ, respectively, represent the mean value and the standard deviation of the response map. A tracking result is regarded as responsible when the *PSR* is above 10 [20], and thus the long-term and short-term filter memory models are updated; otherwise, the target is occluded and then further face recognition is started.

However, most researches merely take the response map of the target in the frame justly tracked into account, in other words, the influence of the maps in the previous frames are neglected.

To be specific, parameters, like *PAR* and *APCE*, etc., vary with different trends during different periods'—usually the variation goes faster when the target is being gradually occluded or it moves away from the occluding background object during the period of partial occlusion. So as to capture the process of the variation of the response map under partial occlusion, the parameter *FCDS* is proposed in [21] to learn the variation feature of the *APCE* in all past frames for the identification of occlusion or background drift, which is formulated as in (4)

$$FDCS = \frac{\text{mean}(\max_N(APCE[0:n])) - APCE_t}{\text{mean}(\max_N(APCE[0:n]))}, \tag{4}$$

where $\max_N(APCE[0:n])$ is the largest *N* values of *APCE* in all previous frames and $APCE_t$ is the value in frame *t*. The correlation filter is regarded as not so reliable when its *FDCS*, namely $FDCS_{cf}$, reaches a threshold, thus an update of the filter tracker and the color tracker is stopped. Otherwise, the two trackers are respectively updated according to their discrimination scores.

2.3. Update Using Similarity Measurement

Multiple-template models are usually adopted in generative models, i.e., sparse coding, in which template sets are updated along with the appearance variation of targets-in usual cases, a target appearance model is updated when the appearance of the tracking result is similar enough to the templates, while it needs to be updated when the similarity is not too low but relatively lower than the normal value, which indicates an apparent appearance change. Commonly adopted similarity measurements are cosine similarity, L1 norm, Euclidean distance, etc.

A template set can well represent a tracking result if the similarity values between it and the majority of candidates are high enough; therefore, it needs to not be updated for calculation reduction, while drift might occur when the similarity falls below a degree. Cosine similarity [22] is used for measurement of the similarity between the tracking result and the templates, where the template with a low similarity value is replaced by the tracked object when the similarity value is between 0.65 and 0.85 to avoid excessive mixture of background pixels. Similar update mechanism is utilized by the extreme learning machine (ELM) framework in [23]; however, the ELM model need not be updated only when the similarity is above the threshold, since the semi-supervised learning mode of ELM model and its strong discrimination ability guarantees the quality of the tracked targets. Soft cosine similarity [24] is defined for the measurement rather than conventional cosine similarity to cope with combined challenging factors, i.e., out-of-plane rotation and apparent scale change simultaneously occur during a period. In [24], a tracking result is departed into several parts, anyone of which does not contain too many background pixels when the soft cosine similarity between it and the corresponding template is no less than a predefined value, therefore that template part is updated in a linear interpolation way, otherwise the update is prohibited. Of the multiple-feature pattern, the absolute error gets lower as the similarity between the specific feature template and the corresponding feature of the tracked target goes larger. The sum value of L1 norm of the subtraction matrix of all the template features and the tracking result is used to reflect the total difference, which is greater than a certain threshold when some of the features have undergone evident variance to measure the difference between the result and the templates. The feature template with the smallest weight is then updated to adjust to the change of this feature of the target.

2.4. A Conservative Updating Strategy

Usually, the reliability of the tracked object needs to be estimated no matter how frequently the model is updated in the regular cases. However, drifts may occur when the surrounding patches that are similar to the object are mistakenly identified as foreground, incurring fatal impacts in the consequent frames if the errors are not erased in time. Under this situation, it is sometimes hard to discriminate true appearance change and occlusion when the difference between the tracked target and the template gets bigger.

A conservative update strategy is proposed in [25], in which the reliability of the tracked object is not considered, to reduce impacts of drifts under background interference. During tracking, a whole sequence is departed into several long time periods, each of which is further divided into smaller ones, and several rather than one trackers are established, of which the amount is equal to the amount of small time periods within each larger ones. Each tracker is distributed with a specific update policy, but the public update must be performed frame-by-frame in the first small period of each big ones, thus each tracker stops updating after a certain amount of small periods and then restarts.

The beginning of the next big period is shown in Figure 1. The tracking framework in [25] is named MT.

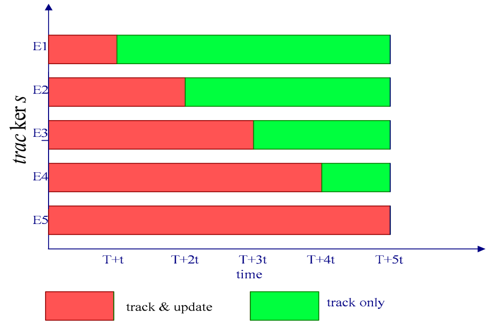

Figure 1. Update Policy of MT.

At the end of a big period, each tracker might track to a position different with which tracked by other ones-some trackers are able to correctly capture the target, while others might fail; therefore, how to select an optimal tracker needs to be further considered. So as to measure the trackers' performances, each one of them tracks the object backwards from the terminal position for a long period equaling to which in the forward tracking, of which the update policy is also the same as in the forward tracking stage. Trajectories of the forward and backward tracking of each tracker are both recorded after the entire process. For a tracker with better performance, the distance between the trajectories of the two different directions is usually comparably lower than others, thus the tracker with the least distance in a big period is selected as optimal.

The tracker that is composed of feature-specific ones named MTM is designed on the basis of the single-feature tracker model named MTS when considering that different features can also bring about different influences to the tracking effect, thus the total amount of trackers equals the product of the amount of features and that of the small periods. The optimal tracker is chosen from all those ones after a round of forward and backward tracking process.

2.5. Combination of Long and Short Term Update

For trackers in many researches, the target model is also updated when a sudden appearance variation or occlusion occurs in addition to when the scheduled update time is up in order to resist drifts that are caused by abrupt target appearance changes or partial occlusion brought about from fixed-time-interval update. It is called update in combination of long and short terms (LST).

For the resistance of impacts of scale variation, deformation, and some other sudden factors, "semantic segmentation" mechanism is introduced in [26], where the correlative parameters of HoG and RGB feature maps between target-based "segmentation map" and position-based "tracking map" are respectively calculated. As long as the target state suddenly changes, the correlation parameter between the "segmentation map" and the hybrid feature map goes higher than that between the "tracking map" and the hybrid feature map, thus the "segmentation signal" is comparably more reliable than the "tracking signal". An immediate target model update is needed to satisfy the real-time changes in this case. Unlike conventional fixed-time-interval update, in this research a frame is regarded as a key frame when the tracking result is judged to be reliable, hence the tracking network is updated when the number of key frames reaches a certain amount rather than frame of a specific index is reached. To avoid erroneous update aroused from occlusion or background clutter, a kind of drift and occlusion detection method is proposed in [27], in which the target model and dual network model are short-term updated while using the best latest tracking results; In addition, the long-term update is performed every ten frames. For adequate use of earlier target information, the score of a tracked target is calculated in [28], which is above 0.5 if the result is regarded as responsible, thus the frame number is added into both the long-term frame number queue (contains 100 frames for most) and the short-term frame number queue (contains 20 frames for most). Appearance variation is detected when the positive classification score is less than 0.5; hence, positive samples from the frames in the short-term queue are used for the network update to meet the demand of pacing with the instant variation. The long-term update is also performed every constant ten frames, when the positive samples from the long-term queue that are rich of previous target information are selected to update the network.

2.6. Module Summary

This module discusses commonly utilized model update occasions. Basic update occasions are listed at the beginning and limitations about time-scale-based update method are briefed next. Recently adopted update occasion determination methods are then illustrated in detail that three kinds of tools for measurement of target's responsibility—occlusion detection, response map, and similarity measurement—and two kinds of newly-proposed hybrid updated occasions—the so-called MT with a conservative update mechanism and LST are respectively illustrated. A reliability check of the tracked object ahead of track can well prohibit erroneous update of the target and tracker model. Additionally, the mixture of long and short term update that fuses the advantages of different update occasions further enhances the adaptability of the trackers. Further solutions to disturbance of similar objects in the target's surrounding area are required in future researches. According to this problem, response check on surrounding background regions should be utilized for the recognition of the true target-the real position can be obtained by comparison of the similarity between the characters of the surrounding background and which of the surroundings templates or utilizing the response maps of the surroundings patches, which might help to alleviate background drifts.

3. Review on Target Model Update Strategies

The design of the model update strategy is a hard project in the work of target tracking. The strong abilities to discriminate the foreground and the background and recover the target after temporary disappearance are not the only requirements for a robust tracker, lower time, and memory consumption as well as an excellent data structure are also essential demands of a good update strategy. In recent years, increasing researches on object tracking have focused on how to balance the robustness of a tracker and low expense of time and memory space. Updated strategies that are based on four commonly-used tracking frameworks—correlation filter (CF), sparse coding (SC), bag-of-words (BoW), and neural network are respectively illustrated below.

3.1. Update Strategy Based on Correlation Filter

Correlation filter (CF) has become one of the most popular utilized models for moving target tracking, especially since Kernelized Correlation Filter (KCF) was first proposed in 2015, and nowadays a large number of researchers have paid attention to the design of filter models with much higher speed, owing to the character of fastness, preciseness, and low expense of time and memory space. Improvement measures of CF model update are also proposed in recent years, having created great breakthroughs over the traditional CF model update method.

Traditional CF target and parameter model update is the linear interpolation of the previous model and the model just trained by the samples from the current frame, as in (5) and (6), which respectively formulates the update of the target model and the parameter model

$$x_t^* = (1-\alpha)x_{t-1} + \alpha x_t\ , \tag{5}$$

$$A_t^* = (1-\mu)A_{t-1} + \mu A_t\ , \tag{6}$$

where x_t and A_t respectively represents the tracking result in the current frame and the tracker parameters, α and μ respectively means the learning rate of the appearance and parameter model. Constant learning rate is widely used in early models [15,29–33]; however, fixed learning rate cannot properly reflect the real variation of the target appearance, owing to the uncertainty of target variation. If the rate remains high when the target is occluded, some background characters will unavoidably mix into the appearance model; otherwise, if it remains a lower value, the target model will not be able to catch up with faster variations of the target [8,28]. Most recent researches have adopted adaptive learning rates that are adjusted to the extent of target appearance variation and the reliability of the tracking result, which increases the robustness of the tracker model to a great extent, in order to avoid drawbacks of the constant learning rate.

In the last two years, response maps are widely utilized to measure the reliability of the tracking results, of which the simplest method is to use the maximum value. A parameter in [34] is defined to adjust the learning rate according to the response of the tracked target, which is equal to the ratio of the maximum value of the response map in current frame to the maximum of all the response values in previous frames in order to avoid impacts aroused from drastic target appearance variations led by background drift, as formulated in (7)

$$\mu = \frac{\mathrm{F}(t)}{\max\{\mathrm{F}(i)\}_{\mathrm{i}=1}^{t\text{-}1}}\ , \tag{7}$$

in which F(t) denotes the maximum value of the response map in frame t; μ gets smaller when improper background drift or heavy occlusion happens, so as to prevent the template model from being contaminated by the tracking result in current frame. The target appearance model is updated as (8), where γ_{init} is the initial learning rate.

$$\hat{\mathbf{x}}_t = \mu\gamma_{init}\mathbf{x}_t + (1-\mu\gamma_{init})\,\hat{\mathbf{x}}_{t-1}\ , \tag{8}$$

Owing to the fact that target appearance varies in a continuous form, the variation remains stable as time goes on in normal situations; hence, response maps in each frame of a sequence are not independent, especially relevant between two adjacent frames. The reliability parameter (denoted as S_t in (9)) is defined in [35], which is the product of negative exponent of the distance between the target center in the adjacent frames and the PSR value in the current frame, to more effectively represent the stability of the appearance variation of a tracking result. Additionally, to put the temporal stability into consideration, previous movement information is further assembled and an increasing sequence $\mathrm{W} = \{\theta^0, \theta^1, \ldots, \theta^{\Delta t-1}\}$, ($\theta$ > 1) is introduced for providing the latest scores with more weights. The learning rate keeps unchanged when the value in the current frame is above μ (is set to 0.7 in the experiment) time of the weighted average of it in the last Δt (=5) frames; otherwise, it decays to the ratio of the reliability value in the current frame to ì time of the weighted average of it in the last five frames, as in ((9), (10), and (13))

$$S_t = \exp(-\frac{1}{\sigma^2}\|C(b^t) - C(b^{t-1})\|^2) \times PSR^t\ , \tag{9}$$

$$\bar{S} = \frac{1}{\Delta t}\sum_i \omega^{\mathrm{i}} S^i\ , \tag{10}$$

$$A_t = (1-\eta)\ A_{t-1} + \eta A_t^*\ , \tag{11}$$

$$x_t = (1-\eta)\ x_{t-1} + \eta x_t^*\ , \tag{12}$$

where $C(b^t)$ denotes the center of the tracked target in frame t PSR^t is the PSR value that is introduced in the second module of Section 2; ω^i is the weight in frame *i*, where the index $i \in [t - \Delta t + 1, t]$, and $\omega^{\mathrm{i}} = \theta^{\mathrm{i}} / (\sum_i \theta^i)$, θ^{i} is the ($i - t + \Delta\ t$)-th element in the sequence *W*; $\bar{S}$ is the weighted average reliability of the last Δt frames. (11) and (12) are, respectively, the formulation of parameter and target appearance model update, η is the adaptive learning rate, which can be formulated as in (13)

$$\eta = \begin{cases} \eta_{init} & S^t > \mu \bar{S} \\ \eta_{init} [S^t / (\mu \bar{S})^{\beta}] & other \end{cases}, \tag{13}$$

where μ is the fixed parameter that equals to 0.7 and β is the decay factor. This update strategy works well during the process of partial occlusion—when the target is being gradually occluded, the size of its visible part is getting smaller. The shape of the response map become increasingly irregular and the target response value goes lower correspondingly; therefore, the reliability value S_t also drops, and the learning rate is adapted lower to avoid improper update (as the lower formulation in (13) when $S^t \leq \mu S$). For the other case, when the target is leaving off the occluding background, the size of the visible part continuously grows, and the response map gradually recovers to the normal shape, thus the reliability value S_t increases. However, the learning rate remains unchanged in this period to inhibit the excessive integration of new target characters that cuts down the universal usage of the model (as the upper formulation in (13) when $S^t > \mu \bar{S}$).

The decrease of response parameters might not be only related to the interference of pseudo targets, self-variation of the target appearance can also bring about the temporary drop in the current and last few frames. The target model is badly in need of an instant update at the moment but it might be disabled if this decrease is mistakenly regarded as the consequence of unreliable variation. The authors in [36] believe that the variation of the target is proportional to its instant speed. Hence, dynamic update of the target model should also be paced with the variation of the speed of the target in addition to the changes of its characters. The learning rate in [36] is determined by two aspects—target moving speed and its feature variation. To get over the problem of partial occlusion that makes it difficult to update, as well as avoid the defect of the speed measurement by distance description, it is believed that the variation of target speed and appearance features are complementary; therefore, the learning rates that are relevant to them ought to be respectively defined, i.e., θ_1 and θ_2 respectively in (14) and (15), which increases with the speed and similarity between the template and the tracked target, respectively. The final learning rate is formulated, as in (16)

$$\theta_1 = \frac{1}{1 + \left(\frac{6}{1+v} \right)^5}, \tag{14}$$

$$\theta_2 = \frac{1}{2} \frac{e^{5c - \frac{5}{2}} - e^{\frac{5}{2} - 5c}}{e^{5c - \frac{5}{2}} + e^{\frac{5}{2} - 5c}}, \tag{15}$$

$$\theta = \alpha \theta_1 + \beta \theta_2, \tag{16}$$

in which *v* and *c* denotes the speed that is measured by the distance between target centers in two adjacent frames and the similarity between the tracking result and the template, respectively; e is the natural exponent base; α and β respectively denotes the adaptive coefficients of θ_1 and θ_2. To learn more about the derivation of θ_1 and θ_2, please refer to [36] for more detail.

The linear interpolation update calculation makes the model sustain the old target appearance as well as introduce new appearance features. The single template model is not able to adequately reflect historical target appearances, although the learning rate can be real-time adjusted according to the response of the target. To overcome this limitation, multiple-template structure, which is being more commonly adopted in generative models, is utilized in some CF trackers, as in [8], to get over the difficulty of calculating the learning rate. Two sets of templates $\mathbf{H}_f^* = \{H_i^*\}_{i=1}^n$ and

$\mathbf{H}_s^* = \{H_i^*\}_{i=1}^n$ are established respectively for the first and second tracking in [8], the former of which is asserted by the tracking result X_t in each frame, i.e., $\mathbf{H}_f^* = \mathbf{H}_f^* \cup \{H_t\}$, $H_t = G / X_t$, where G denotes the trained filter parameter image. In the meantime, a template with a relatively larger difference from the result and lower confidence value is removed from the set. Similar to the representation form of sparse coding, the tracking result is linearly represented by the target template set $\mathbf{F} = \{f_i\}_{i=1}^n$

$$f_t \approx \mathbf{Fa} = \sum_i f_i a_i \,, \tag{17}$$

in which the coefficient vector $\boldsymbol{a}$ can be solved through sparse coding and it is used for the generation of candidate regions for the first track in the next frame. The second track template is acquired by the combination of the first track template and the original template in the next frame, which is used for the selection of the optimal candidate as the tracking result. The formulation of the second track template is as in (18)

$$\mathbf{H}_{si}^* = (1-p)H_t^* + p\mathbf{H}_{fi}^* \,, \tag{18}$$

where p is the proportion parameter.

A multiple-filter template structure is adopted in [37] to form a strong CF classifier based on the CFs from current and previous frames in order to utilize historical parameter models. To reduce calculation complexity and memory consuption led by storing similar CFs from adjacent frames, CFs are clustered. After the target in frame n is tracked, the CFs in the last r frames, including #*n*, are firstly added into the CF set while those in other *n*-*r* frames are clustered into K classes; afterwards, the CF with the lowest classification error in each cluster is added into the CF set. The K + r CFs are combined with different weights to form the final strong CF, which can be formulated as in (19), and ρ_{n}^i is the weight of the *i*-th filter in frame #*n* calculated as in (20), where e_i denotes the training error of the filter calculated as in (21), in which (x_t, y_t) is the new training sample of the *t*-th frame, whose spatial size is *M*×*N* and each sample $x_t^{(k)}$ of x_t is a d-dimensional vector $[x_t^{(k)(i)}]_{i=1}^l$ $\hat{x}_t$ and $\hat{f}_t$ are Discrete Fourier Transforms (DFT) of x_t and f_t and w_k denotes the weights of all samples $x_t^{(k)}$, which is defined as in (22). However, the CF set is updated every certain frames rather than in each frame to cut down calculation burden and prevent useless operations.

$$f_n^{strong} = \sum_{i=1}^{K+r} \rho_{\mathrm{n}}^i f_n^i \,, \tag{19}$$

$$\rho_{\mathrm{n}}^i = \frac{1}{2} \ln\left(\frac{1-e_i}{e_i}\right), \tag{20}$$

$$e_i = \sum_{k=1}^{M\times N} w_k \left(F^{-1}\left\{ \sum_{l=1}^{d} \hat{x}_n^{(k)(l)} \bullet \hat{f}_n^{i(k)(l)} \right\} - y_n^{(k)} \right)^2 , \tag{21}$$

$$w_k = \begin{cases} Y_k / \sum_{m,n} \exp\left(-\sigma\left((\mathrm{m} - M/2)^2 + (n - N/2)^2\right)\right) & \textit{at the beginning} \\ \frac{w_k}{\sum_k w_k} \exp\left[\rho_{\mathrm{n}}^i \left(F^{-1}\left\{ \sum_{l=1}^{d} \hat{x}_n^{(k)(l)} \bullet \hat{f}_n^{i(k)(l)} \right\} - y_t^{(k)} \right)^2 \right] & \textit{others,} \end{cases} \tag{22}$$

3.2. Update Strategy Based on Dictionary Learning and Sparse Coding

Dictionary learning (DL) and sparse coding (SC) are common generative frameworks of visual tracking. The template set is usually made up of the tracking results from each frame, while at the beginning stage of tracking it consists of the positive and negative samples drawn in the first frame. Two common ways generates the dictionary [38], one of which is through learning methods, i.e., principal component analysis (PCA), where the dictionary is acquired by the form of iterative training of samples in specific frames, the other is to directly insert the tracking result into the template set and then select a subset. The latter method is more popularly adopted in the recent year.

The dictionary model needs to be constantly constructed with the appearance variation of the target and background. When considering that there are some slight differences between two adjacent frame images, from each frame positive and negative samples should be added into the sample set. However, there are at least two variables must be iteratively solved in the normal dictionary learning framework—the dictionary and the sparse coefficients, obviously calculation burden will increase if the dictionary is updated every frame that unnecessary updates may have consumed a lot of time. For the balance of tracking accuracy and efficiency, in [39], foreground and background samples are preserved after tracking in each frame, but the dictionary is updated every *T* (=15) frames, which is mainly trained while using the target and background samples in the last 15 frames and is emptied whenever the dictionary update is finished. Target samples in the first frame and the sample that is calculated as the mean image of all the best tracked results are also used for training and never deleted after updates to overcome the impacts of bad positive samples arouse from occlusion or background drifts. Similar dictionary learning way is utilized in [40], whereas background samples are not used for dictionary training, and the method in [40] is the improvement of the space sparse learning (SSL), which fixes too much attention to positive samples in the latest frames while ignoring the contributions of distant tracked frames, which might unavoidably make the template integrate with too many newest characters that makes the tracker hard to re-detect the target after full occlusion or out-of-view.

Currently, the latter dictionary construction method that selects a set of reliable tracking results as the dictionary has been more popularly utilized, which is termed as sparse coding, in order to cut down the calculation burden brought about by dictionary training and alleviate the impact arouse from irregular sample distribution generated from fixed-time-interval update. The simplest way is to directly use the tracking result in the current frame as the new template and insert it into the set or replace one with the least similarity in the set with it. However, owing to the reality that the image of a tracked object is often interfered by pseudo target pixels or noises aroused from irregular illumination, the target model might get distorted if the raw tracking result is directly added to the set. To eliminate the influence of noises, trivial templates [41,42] are usually used for target image representation, which is expressed as in (23)

$$\min_{\mathrm{z}} \| \mathrm{g} - \mathbf{B}\mathrm{z} \|_2^2 + k \| \mathrm{z} \|_1 \text{, s.t. } \mathbf{B} = [\mathbf{E}, \mathbf{I}], \mathrm{z} = [\mathrm{a}', \mathrm{h}'], \tag{23}$$

in which g denotes the raw target image, **B** is the template set that is composed of a denoised template set **E** and a trivial template **I**, and a′ and h′ are their coefficients correspondingly. The denoised target image **T** = **E**a′ rather than the raw image is used to update the template model. So as to overcome the defect of less enough contribution of the denoised templates due to the excessive sparsity effect on them, the sparsity constraint is only imposed on the trivial template set in [42], which is formulated as in (24)

$$\min_{\mathbf{q},\mathbf{e}} \| \mathrm{p} - \mathbf{U}\mathrm{q} - \mathrm{e} \|_2^2 + k \| \mathrm{e} \|_1, \tag{24}$$

where p is the image of the raw tracking result, q and e are respectively the coefficients of the denoised template set and the trivial template set, and **U** is the eigenbasis of p. The final image $\tilde{\mathrm{p}} = \mathbf{U}\mathrm{q}$ is inserted into the template set. Although the template set is also updated every a few (=5) frames, to make it more representative that it should not contain too much newest characters or too old ones, the set composed of 10 templates is established, where the target in the first frame is permanently preserved in the first room, while tracking results are stored in other nine rooms in time order. The templates in room 2, 5, and 8 are removed and the denoised results in three editions are added at the rear.

A global update of templates makes the model less complicated and the calculation burden is thus alleviated. However, the representation of each target part should not be the same due to the truth that different features are contained in different regions of a target image. Besides, the sparsity constraint does not work well if a template set that can only globally represent image is used. When considering different variation form of each part and the effect of partial occlusion, target dictionaries are not only the subset of a template set according to the theories in newest researches, patch dictionaries are usually established instead of holistic ones that a specific region of the templates are used to construct the local dictionary of that target region [16,17,41,42]; therefore, different update policies are utilized on different local patches. For more robust representation of visible object parts when other parts of the object are occluded, the object is represented in a different form from the situation of no occlusion in [16]. During partial occlusion, the contribution of each template patch is calculated while using the tracking result—occluded patches contribute much less to the representation, therefore template patches with a higher contribution value can be effectively updated while the update of other patches is temporarily prohibited. To eliminate the impact of background pixels in a target image and make the tracker model more robust to deformation and rotation, object patches are classified into three types: stable patches, valid ones, and invalid ones, and three types of dictionaries, called total dictionary, object dictionary, and background dictionary are constructed in [17], which has been illustrated in the first subsection of Module 2. The target dictionary $\mathbf{D}_o$ is updated while using valid patches.

3.3. *Update Strategy Based on Bag-of-Words*

Objects in each frame of a tracking sequence can be only classified into two classes—object and background. In terms of animal's vision mechanism, the classification of two different types of objects is usually according to the characters that are not the same among them, which gives the inspiration of bag-of-words (BoW) model in the domain of visual tracking, for the fact that in general characters contained in the foreground are distinguished from that in the background, thus there should be plenty of symbolic features to assist in object classification. However, there have not been too many tracking algorithms that are based on this framework when compared to other ones up to now, and less robustness has been shown in the tracking performances, for the reason that most of them neglect the consideration of the holistic structure of the target and background.

Visual "words" are the visual characters from the area of the target and background in a tracking frame that are used as training samples in discriminative frameworks. For instance, in [42], the "words" are classified in a supervised way while using SVM. During the update process, new visual foreground "words" and background ones are extracted from the region of the object and a random background patch, respectively.

However, the background and foreground in one frame might share some "words" with similar features, therefore a background character might be mistakenly classified as a target if it is much too similar to some features in the target feature bag. Hence, the target "words" like these cannot be used for discrimination. In [7], the authors believe that target occlusion might well happen when there exist features in the bounding box that are similar to or even the same as those in the surrounding area. If the number of these features is larger than usual, occlusion can be surely regarded to have occurred. In usual condition where no occlusion happens, foreground and background features in the bounding box are respectively merged into the target feature set and the background one; afterwards, other background features are searched from the surrounding background in the past few frames and then merged into the new background feature set, which has made the background more distinguishable that false targets have lower probability to be misidentified as the true one. A similar unsupervised way is utilized in [43], in which if the distance between a word v_i in the context bag M_B^t and its nearest neighbor word v_n from the bag of the last frame is lower than the threshold τ_B, a new word v_{new} in combination of the two words is added into the word bag in the current frame, as in (25), where C denotes the flag of background or object and $\acute{a}$ is the proportion parameter; otherwise, when the background word bag M_B^t is updated,

v_i is directly merged into the bag: $M_t^B = M_t^B \cup v_i$. If it is time to update the object word bag M_t^O, there is a need to check whether the current word v_i is reliable, which is measured by the distance between it and its neighbor word v_m from the newly updated background word bag M_B^t and that between it and the neighbor v_n from object bag M_{t-1}^O of the last frame. The word is regarded as reliable if the latter distance $d(v_i, v_n)$ is smaller than the former, named $d(v_i, v_m)$; thus, it is merged into the object word bag in the current frame: $M_t^O = M_t^O \cup v_i$; otherwise, no bag is expanded. In addition, when any of the two bags is full, some words are randomly removed from the bag.

$$v_{new} = (1.0 - \alpha)v_n + \alpha v_i\,,$$
$$M_t^C = M_t^C \cup v_{new}\,, \tag{25}$$

3.4. Update Strategy of Neural Network Models

A series of neural network framework have been widely adopted in researches of visual tracking because of its strong capability of feature extraction and image classification, of which researches on the improvement of accuracy, speed, as well as the structural layouts are gaining rapid progress. Quantities of labeled images are used for iterative training and during training features of different depths that describe the trained samples from different aspects are extracted, thus a set of parameters with high validity are finally determined thanks to the neural structure of it, which greatly alleviates the tedious process of handcrafted feature extraction in traditional machine learning models. A huge challenge of visual tracking under neural network framework today lies in the shortage of training samples as well as in the sensitivity to irregular sample distribution and noisy samples [44], of which the sample distribution and quality of training samples decides the capacity of a network to a large extent. So as to further boost the capacity of tracking networks, the hot topic of tracking under deep neural network has recently transferred to the further procession of training samples, which is a credible mark of progress in the research of deep learning.

The distribution of foreground and background stays stable during tracking in a short period. The samples used for model update should possess two characteristic to make the network adjust to the appearance change of the target: firstly, the frames that the positive samples are selected from should be as close as possible to current frame to ensure the real-time requirement; secondly, it must contain a correctly tracked object that is without the influence of occlusion or drift. In other words, it must be responsible enough. Based on these two characteristics, during the stochastic (short term) update reliable samples are picked out for model retraining in [27]. To make the target model less dependent on newest appearances and cope with the lack of positive samples when temporary target loss occurs during periodically (long term) update, positive samples from the first frame are also used for the update as supplement in addition to from the best tracked frames. The similar method that takes the samples in the first frame into account is also utilized in [45], where Gaussian maps of each frame image also take part in the update training.

The initial appearance is preserved in a network model if the target samples drawn from the first frame are put into consideration when updated, which is helpful for target re-detection after its reappearance after temporary disappearance. Pessimistically believed in [46], from the author's point of view, only the positive sample from the first frame is completely reliable, whereas contamination and decision mistake must exist in other frames to some extent, which is also deemed to be true in [47] that there must exist error a bit or too much in each frame, except in the first one. However, optimistically speaking, thanks to the close appearances from the two adjacent frames, a trend of the variation can be foreseen within a small period (no above than three frames); therefore, there exists a high confidence of making sure whether the tracking result is responsible. As a matter of fact, the target appearance might have undergone variances plenty of times after hundreds of frames of tracking, it is not sufficient to achieve re-detection only through the target appearance in the first frame; since, in usual cases, the real appearances of the target in the last few frames are much closer to that in the re-detection frame, as the assumption that target samples that satisfy the two conditions listed in last paragraph should be more important. Target reliability detection is utilized in some researches so as to use more reliable samples, whereas the best-fitted positive samples are selected for retraining. A read-and-write memory structure is established in [46], to which the tracked object is inserted and the sample with the lowest confidence is removed from it unless it is full. During the update, scores of importance are given to the selected samples from the memory for calculation of the gradient descent parameters. For adequate use of the reliable samples in the past frames, the self-paced selection model is adopted in [48] to control the selection of positive samples, those with the lowest loss value based on the current loss function are selected for network retraining, and the criterion of the samples to be selected for retraining in a frame is based on the overall reliability of the samples in the previous frame.

3.5. Module Summary

In this module, the update strategies under four mainstream tracking frameworks—correlation filter, sparse coding (dictionary learning), bag-of-words (features), and deep neural network—are discussed, and the progresses are illustrated according to the specific examples in recent researches. Questions regarding the challenges that remain in the existed update strategies are summarized as below: (1) How to build a template set structure that includes more abundant information about the target but consumes the least amount of memory as possible; (2) How to more effectively choose training samples that contain various kind of target appearances and control the distribution of the sample set for deep neural network update; (3) How to deal with visible parts of the occluded target and make good use of them for update to boost the network's adaptation to newest appearances; and, (4) How to separately use different features of the target and utilize feature-specific update methods to make the tracker more robustly adjust to the variation of each feature. Contributions of each feature or convolution layer should also be considered for the update at the global level.

4. Background Model Update

The environment of the target existence is background. With the movement of the target, the background also varies its appearance, so the correct estimation of targets' surrounding background is the premise of correct location of the target. Characters of the background regions that surround the target are especially essential to prevent drift to similar objects in the background, which should be distinguished from the characters of the target [49]. Compared with the target, the background occupies much larger area in the view of a frame, whose appearance features appears more complicated, hence there ought to be plenty of available negative sample sources, therefore how to more credibly select background samples is also a key part in the work of update. Background model update occasion and strategies are discussed below.

4.1. Background Sampling Methods

Sampling of background samples is the key part of the update work, owing to the fact that the number of background patches is far larger than that of foreground patches. Random selection is adopted in some researches, for instance, background "words" are extracted from random regions outside the target area in [31]. Yet, an object must exist in a specific environment—it must possess an exact position in the background area. Based on this truth, the authors in [2] hold the view that all non-overlapped background patches are not equal, and background regions with different features affect apparently differently on the sample classification. In this research, sampled background patches are clustered into multiple groups; afterwards, the specific SVMs are trained using each group of the background and target samples. Negative samples distant from the target area are drawn for update to make the foreground samples more distinguishable, where the sampling method is often utilized in extreme learning machine (ELM) [3,50] frameworks. Some SC based trackers also use background patches faraway from the object, i.e., [41,51].

Nevertheless, not all of the background characters are of valuable use. On the basis of animal's visual tracking mechanism, the background regions near to the area of the moving object contain the most valuable information that can help with target location; hence, they ought to be the most available parts through the entire background, while the influence of the information of background far from the position of target are of far less importance. The examples of background sample selection policies in last paragraph overlook the relationships between the target and its context, which violates the mechanism of animal's selective attention, despite the fact that the ELM frameworks are robust enough to fight against the diversity of sample appearances. Luckily, there are an increasing number of researchers who have realized that mechanism that background characters close to the target area ought to be given the highest importance. For instance, background samples that are drawn near to the target region are used for the dictionary model update in [52], which is the spatial constraint of the data sampling in the article, in which the temporal constraint is that the samples selected for training should be from the latest few frames. This distance constraint is also satisfied in [46] by the update of the network model. Of the bag-of-words (features) based tracking frameworks, as in [7] and [43], words or features in the surroundings near to the target are used to update the context (background) bags when the foreground bags are usually updated in parallel, which has been illustrated in detail in the third subsection of Module 3. The parameter of intersection over union (IoU) is usually used to identify whether the patch that is selected around the target is foreground, the patch is regarded as a positive sample when which is above a higher threshold, or a negative one if below a lower threshold. In [28] and [53], IoU is used to help draw positive and negative samples for network update. Samples whose IoU are between the two thresholds are also picked out for network retraining in order to increase the robustness of target position and make abundant use of visible parts of a tracked target when partial occlusions occur.

Dense sampling is commonly utilized as for the density of sampling, like some particle filter based sampling methods, i.e., [51]. Dense sampling means that positive and negative samples are drawn within a length of radius according to a given distribution, i.e., Gaussian Distribution, in which there is a large overlap between any two of the samples of the same class. The advantage of this kind of sampling approach lies in that it not only makes abundant use of the background information around the target thus strengthen the discrimination capability of the tracker, but it also helps to provide more sufficient source of samples, which boosts the robustness of deep networks.

4.2. A Kind of Background Unity Estimation Approach: TBE

Up to now, most tracking algorithms have concentrated a lot on the construction and update of target models, while those of background models have been rarely researched. The distribution of the feature of the target's surrounding area is usually irregular, owing to the complexity of the background. Therefore, the requirement of accurate target location cannot only be satisfied through simple target matching methods. When the target is occluded, its appearance has gotten incomplete that available target characters have become less, which makes it hard to distinguish from the background. An original method, named Tracking by Background Estimation (TBE), is proposed in [12], which includes the approach of background modeling and update strategy by which foreground pixels are extracted out for target detection and location, to achieve more accurate target location especially in the state of partial occlusion.

TBE is based on the principle of background subtraction, through which the preserved area of foreground pixels is used for target detection and location; afterwards, the appearance model of the target is learned. Suppose that the entire image f_i is composed of a target t_i and a background b_i i.e., $\mathrm{f}_i = \{\mathrm{t}_i, \mathrm{b}_i\}$, where *i* is the frame index, and the mask of the background b_i in frame *i* is identified as m_i. All the pixels in the image domain of f_i compose the set P_i . Given a pixel $\mathrm{x} \in \mathrm{P}_i$, if x belongs to the background, there is $\mathrm{b}_i(\mathrm{x}) = \mathrm{f}_i(\mathrm{x})$ and $\mathrm{m}_i(\mathrm{x}) = 1$; otherwise, $\mathrm{b}_i(\mathrm{x}) = 0$ and $\mathrm{m}_i(\mathrm{x}) = 0$. To eliminate the influence of background illumination, "mean-background" is defined and suppose $\tilde{\mathrm{b}}_\mathrm{i}$ is the mean-background in frame *i*, the corresponding mask of which is $\tilde{\mathrm{m}}_\mathrm{i}$. All of the pixels in the image domain of $\tilde{\mathrm{b}}_i$ compose the set $\tilde{\mathrm{P}}_i$.

Assume that the camera is stationary, the background in two adjacent frames is completely the same, thus $\mathrm{t}_i = \mathrm{f}_i - \mathrm{b}_{i-1}$, and the target can be recognized by means of the subtraction of the frame images. Yet, in almost all cases, the camera is in movement sometimes, which brings about the deformation and scale variation of the background. Based on this factor, the warped image in frame *i* is identified as $\hat{b}_i$, which is transformed from the mean-background in the last frame, as in (26)

$$\hat{b}_i = H_i * \tilde{b}_{i-1}, \tag{26}$$

where * is the transform operator and H_i is the calculated homography matrix. The warped $\hat{b}_i$ from the mean-background in frame *i*−1 suits to the background in the current frame *i*, making the background subtraction applicable. Thus, the mean-background in frame *i* is calculated by the weighted sum of $\hat{b}_i$ and b_i , as formulated in (27)

$$\tilde{b}_i(x) = w_i^T \bullet (\hat{b}_i(x), b_i(x)) \text{, s.t. } x \in \tilde{P}_{i-1} \cup P_i, \tag{27}$$

Some background regions in the previous frame do not appear in the current frame and new background regions may appear due to the movement of the background. Besides, the target must exist in the shared parts of the background regions, i.e., $x \in \tilde{P}_{i-1} \cap P_i$ if $x \in t_i$; hence, the weight w_i is defined as in (28)

$$w_i = \begin{cases} (1,0)^T & x \in \tilde{P}_{i-1} \wedge x \notin P_i \\ (\hat{m}_i(x), m_i(x)^T)/(\hat{m}_i(x) + m_i(x)) & x \in \tilde{P}_{i-1} \cap P_i \\ (0,1)^T & x \notin \tilde{P}_{i-1} \wedge x \in P_i, \end{cases} \tag{28}$$

in which $\hat{m}_i$ is the warped mask. Subsequently, $\tilde{m}_i$ is calculated as in (29)

$$\tilde{m}_i(x) = \begin{cases} \hat{m}_i(x), & x \in \tilde{P}_{i-1} \wedge x \notin P_i \\ \min\{T, m_{i-1}(x) + \hat{m}_i(x)\}, & x \in \tilde{P}_{i-1} \cap P_i \\ m_i(x), & x \notin \tilde{P}_{i-1} \wedge x \in P_i, \end{cases} \tag{29}$$

where *T* is the predefined threshold that upper bounds the maximum of $m_i(x)$, ensuring the contribution of the latest frame, without which the weight of the mean-background might rise to a high value and the weight of the input frame will be negligible.

An update of the background model is performed after the target is tracked in every frame that the parameters $\tilde{b}_i$ and $\tilde{m}_i$ are obtained and the warping operation is done before target detection in the next frame. Afterwards, background subtraction is conducted for the detection and location of the target.

4.3. Occasions of Background Update

Because variation of the background mainly relies on its movement, though some of its features may passively vary with the environment, it must exist in every frame, the reliability of it should not be given too much consideration, therefore sophisticated discussion regarding the background update occasions is not necessary. The background appearance is temporarily stable thanks to the variety of background patches and the movement along with the target. Usually, fixed-time-interval background update is adopted in SC based and deep neural network based models, and unsupervised models, like BoW (or BoF), update the background model along with the target model frame-by-frame. Negative samples drawn from the latest frames are used for model retraining, which is the guarantee of the requirement of the adaptation of the tracker to the newest background features.

4.4. Module Summary

This module discusses background update strategies and occasions, including a new background model update strategy named TBE. Although the update of background model seems to be much simpler than that of target model, there are still needs of improvements in many respects. The questions remaining about the background update are as below. (1) How to utilize the background information that is useful for discriminating the target and the surroundings for the extraction of key background characters that can help with target location; (2) How to determine the density of background patch sampling. Background regions containing much more valuable information ought to be more densely sampled to boost the efficiency of the tracker; (3) How to build the holistic structure of the background. Algorithms about the background update in the global level should better be designed in future tracking researches, as patches or visual words drawn from background are placed in order in the original image.

5. Analysis on Experimental Results

Challenging factors in visual tracking include occlusion, in-plane and out-of-plane rotation, illumination variation, background clutter, fast motion, abrupt deformation, and scale variation, etc. The robustness of a tracker is measured by its performances under these situations on specific sequences. A successful track means that a tracker is able to track the target without drift through the whole sequence in spite of any of those factors in the video. Whether a tracker can successfully track the target in a sequence depends on the quality of the model update to a large extent. This Section will discuss the tracking experiments from recent researches, where performances under those challenging factors are talked about in detail. The advantages in contrast to the benchmark trackers as well as some failure cases are listed and analysis on the merits and drawbacks with respect to the update strategies are then illustrated. Improvement measures are proposed among the analysis.

5.1. Update Strategies from Recent Researches

Some typical tracker models are listed in this subsection to illustrate the merits and drawbacks of recent trackers, as in Table 1. Table 2 lists abbreviations for the names of the listed frameworks.

Table 1. Model update strategies in recent years and their merits and drawbacks.

Tracker	Framework	Update Strategy	Performances
L3SCM [24]	PF	A local region of the template is updated when the similarity between it and the same region in the target image is no less than a threshold.	Targets can be correctly tracked no matter any challenging factor happens.
MSRBTP [54]	PF	For each feature, the weight is cut down by multiplying a positive value smaller than 1.0 when there is a classification mistake.	Targets can be stably tracked when there are illumination changes. In sequences of *Skiing* and *MotorRolling*, the targets are failed to be detected after presence of scale variation, out-of-plane rotation and out-of-view.
TBE [12]	Background Similarity	Background image model is updated every frame. The target is relocated with the help of the new background model when there is no full occlusion and the target appearance model is updated when no occlusion happens.	Occlusion cases can be correctly identified and the tracker can re-detect the objects after long-time full occlusion. The tracker remains its robustness even if the target constantly changes its appearance, especially when background clutter or out-of-plane rotation happens. However, it is not able to recapture small-sized targets.
ALIEN [7]	BoF	Occlusion does not happen when the number of background features in the bounding box is small, thus target features in the box are transformed and then merged into the foreground feature set while background features in the search regions of the last few frames are merged into the background feature set.	The target can be re-detected in a short time after full occlusion. The tracker is not sensitive to similar objects in the surroundings and is able to accurately measure the size of the target.
ELMAE [3]	ELM	Target template is updated when the distance between the template of the newest result and the template of the target in the first frame is lower than the threshold. Negative samples faraway from the targets are used to update the background model.	Performances on typical sequences that include mixed challenging factors and severe occlusions are much better than benchmark trackers, especially able to deal with the problem of constant rotation in *Freeman1*.
PML [50]	ELM	Positive samples and negative samples far from the target area are selected to update the ELM model, which is performed every certain frames.	The tracker is able to tracker 12 challenging sequences. It is able to accurately detect the target when there is severe in-plane or out-of-plane rotation or scale variation.
SPDCT [48]	DNN	Positive samples with lowest loss values are chosen for network model update every five frames.	The tracker can handle severe problems such as deformation, occlusion and background clutter compared to the benchmarks.
DNCT [45]	DNN	Tracking results in the last six frames and positive samples from the first frame are used for model update when the maximum value of the response map is higher than the threshold.	Targets can be re-detected after full occlusion even if they are much smaller than the normal size.

HCF [29]	CNN+CF	Regular linear interpolation method is adopted during the update of the filters of each CNN layer.	The targets are failed to be tracked in the sequences of *Girl2* and *Lemmin*g; For *Singer2* sequence, the darker foreground is extremely hard to be distinguished from the brighter background for the reason that combined information by multiple layers are used.
DNT [27]	CNN	Short term update is performed using the best tracking results in the latest frames when occlusion or background drift happens while long term update is done using recent results and the target samples in the first frame every certain frames.	Targets' scale and position can be accurately determined even though in the situations of fast motion or background interference.
WALSA [22]	SC	The tracking result is added into the template set and an old template is randomly removed when the similarity value between them is within the range of 0.65 and 0.85.	Targets can be stably tracked under any challenging situations.
TPS [14]	LR	Training approaches of SVMs are applied for the vector regression model SVR. Contribution values of each target part is gotten for the local update.	Strong robustness is displayed in the situations of partial occlusion and deformation.
ODLR [39]	DL	The dictionary which includes background samples is updated every 15 frames, during which positive samples from the first frame and a set of tracking results in the last few frames as well as the mean sample of historical best tracked targets are used, afterwards the set consisting of recently tracked objects is emptied.	Targets can be correctly located whichever any challenging factor comes across. However, for *Pedestrain2* during the reappearance after the disappearance of the walker, false samples are used to construct the object dictionary. For *Skiing*, the dictionary also failed to be constructed as the size of the target becomes too small.
SALSC [41]	SC	Denoised tracking results are used for update. Three templates in the set are replaced by different forms of the result.	Under various kinds of mixed challenging situations like occlusion + background interference, illumination change + rotation, scale variation + background clutter + rotation + illumination change, etc., the targets are still able to be stably tracked.
approach in [40]	DL	The dictionaries are updated using latest tracking results every 15 frames.	The tracker performs excellently at dealing with newly varied appearances.

CRSRCF [55]	CF+SD	The correlation filter and the weight map of the saliency map are updated in each frame.	Objects with irregular shape and heavily deformed targets can be correctly tracked.
LSHR [33]	CF+CNN	The model is updated each frame. When the distance between the target's exact position and the estimated position is bigger than the threshold the state of the target is recalculated using features extracted by shallower layers.	Targets in more challenging videos can be well accurately tracked, especially for the sequence of *Ironman*, only the proposed tracker is able to track whole of it.
DSARCF [56]	CF+SD	Target feature maps from the first frame to the current are used to update the CF in the next frame. The CF and the spatial weight map are updated every two frames.	Under occlusion or heavy appearance changes, targets can be successfully detected. In *Girl2* sequence, when the girl's face reappears after occluded by a man's face, it can be correctly tracked for quite a long time. Yet the saliency map does not work well in the situation of fast motion. In sequences of *Matric* and *Drangonbaby*, the targets failed to be detected using the saliency maps when low resolution or background clutter occurs.
CLIP [34]	CF+SVM	The learning rate is adjusted according to the ratio of the maximum response value to the sum of which in all previous frames. Image patches with highest SVM classification scores are used for the update of SVM as long as the maximum response value of the synthetic features is above the threshold.	Compared with the benchmarks, the tracker performs with much more robustness no matter any challenging situation occurs.
SRKCF [35]	KCF	The credibility value is calculated by the distance parameter between the tracking results in adjacent frames and the PSR value. The learning rate is sustained if the credibility value in current frame is above the average of which in the past few frames, otherwise it is reduced by the ratio of them.	The proposed tracker has better performances than other compared KCF trackers, typically it outperforms others at occlusion handling. In the sequences with background clutter and deformation like *Basketbal*l and *Bolt2*, only the proposed tracker is able to accurately track the targets.
AECF [37]	multiple CF	The final strong CF is updated every 5 frames. The CFs in last several frames are firstly preserved while others are clustered into many groups, thus one CF is picked out from each cluster. These CFs are combined to generate the strong CF.	For *Skating* where the target reappears, it can be retracked. The tracker is also robust in coping with the background clutter problem in the sequences of *Shaking*, *Panda* and *Dragonbaby*.
OSAMCF [57]	CF	Position CF and scale CF are separately updated, whereas the target model from the first frame is also used to update the former one.	Targets can be stably detected no matter any challenging problem comes across.

HDT [58]	CNN	The regret value of each frame that is the cumulative value of the loss values in all past frames is updated by the distance and the appearance difference to further calculate the weight of each feature. The network is updated incrementally using samples in current frame.	Drifts can be well avoided in the sequence of *Basketbal*l where there exist objects similar to the tracked player in the surroundings.
MLFF [18]	CNN	The model is updated only when the maximum value of the response map and the PAR value are both above the average of the historical values.	The proposed tracker performs superior to FCNT, SiamFC and CF2 under most of challenging situations.
PMC [59]	KCF	The basic (first) classifier is never updated. The first and second classifiers are updated when the scores of them are no less than that of the third classifier and the predefined threshold, while the third classifier is updated when its classification score is above than that of the other two classifiers.	The proposed tracker performs extraordinarily well under the mixed challenging factors of partial occlusion and rotation, i.e., *Girl*.
RDLT [51]	KCF+SC	The CF model is updated unless the HoG and color score are both above the average, meanwhile histograms of the foreground and background as well as the sample templates used for re-detection are also updated.	Targets can be correctly recovered after drift loss. However, for *Face-ce*, due to the high similarity between the character of the occluding object and the tracked object, recovery of the target is failed. Also the algorithm does poorly in handling the fast motion problems in *MotorRolling* and *Bike-ce2*.
LSA [17]	SC+PF	The total dictionary is updated when there are enough stable patches and valid ones. The object dictionary is updated using valid patches while background dictionary is updated using local background patches around the target.	Problems of out-of-plane rotation and illumination variation can be greatly handled and partial occluded targets can be accurately tracked. But the proposed tracker is not able to cope with severe scale changes.
NMC [42]	SC	The background template is updated 5 frames; When the number of the background patches that take part in the representation of the tracking result is no more than one, there is no severe occlusion, thus the target template set is updated.	The proposed tracker performs excellently on many sequences with challenging factors.
CBOD [9]	KCF	The kernelized correlation filter is regularly updated unless no occlusion happens.	The proposed tracker performs excellently under various occlusion situations, i.e., sequences of *Tiger1*, *Coke, Basketball, Football, FaceOCC1, CarScale.*

Table 2. Abbreviations for name of the frameworks.

Abbreviations	Full Name
PF	particle filter
DNN/CNN	deep/convolutional neural network
(K)CF	(kernelized) correlation filter
SC	sparse coding
DL	dictionary learning
BoW/BoF	bag-of-words/bag-of-features
ELM	extreme learning machine
LR	linear regression
SD	saliency detection

5.2. Qualitative Advantage Analysis of Some Trackers' Performance on Typical Sequences

To evaluate the quality of a tracker, its performances under those challenging situations, such as occlusion, in-plane or out-of-plane rotation, etc., are usually the accordance, which essentially depends on the quality of the model update strategy. This subsection gives analysis on specific cases where the performances as well as advantage analysis of the recent trackers under the factors of occlusion, background interference, rotation, scale variation, and deformation are respectively illustrated.

(1) *Occlusion*: Occlusion is a hard problem that almost occurs in all sequences, the update strategy under which situation measures the robustness of a tracker to a maximum degree. In sequences of *Jogging-1* and *Subway*, the walkers are respectively occluded by the telephone pole and other passers-by, only the tracker in TPS [14] and the benchmarks of TGP, SCM, and KCF are able to stably track them, which explains that updating in local patterns has provided assistance in tracking partial occluded objects via visible parts. Local feature representation is adopted in [54], where the global feature pattern is fused with local ones to represent the tracked object. In the sequence of *Walking* when the walking woman reappears after occluded by the man, the tracker in [54] can successfully recapture the woman, while the compared benchmarks, like OAB, MIL, and COM, fail to retrack it. The local-patterned update is also adopted by L3SCM [24], which has gained better performances than the compared benchmarks. SC based LSA [17] shows strong robustness in handling occlusion thanks to the use of stable patches and valid ones for update. In the sequence of *Jogging2*, after the occlusion of the walker by the telephone pole, the compared KCF and DSST fail to cope with the drift problem. Different template patches are used to represent the tracked object by NMC [42], whereas the distribution of foreground and background templates is used for the detection of occlusion, which shows its superiority in occlusion handling in *Suv* and *Jogging2*.

Utilization of background models is the key of correct target localization. The target is completely occluded in the frames #27 to #36 of *Uav*, thanks to the constant utilization of background model in TBE [12], the appearance of the target is preserved before the start of its full occlusion; therefore, it is able to be retracked after it reappears, while other compared trackers fail to re-detect it. In the sequence of *Thuyx*, the characters of the surrounding is similar to that of the target, still only the proposed TBE can correctly track it while drifts to the surroundings occur when using other compared trackers. These cases have given us the inspiration that the background model is typically essential in dealing with occlusions. Bag-of-feature based ALIEN [7] effectively prohibits the drift problem in the sequence of *FaceOCC1* due to the use of the background characters. The background information in the tracking bounding box are used to describe the reliability of the tracking result in [40], thus the target model update is prevented if there is too much background information, so for the sequences where there are partial occlusions, i.e., *Coke*, *Girl*, *Lemming* and *Tiger1*, the tracker performs well.

Valuable use of positive training samples plays an essential role in dealing with target re-detection. In frame #131 of *Girl2,* where the man's head moves away and the girl's head return visible, DSARCF [56] is able to perfectly retrack the girl's head while other trackers fail, due to the reliability check of target training samples that are used for the spatial weight update; PMC [59] also performs well on this sequence, even though the face slightly rotates in the process of being occluded, which can be attributed to the complementary update policy of the three classifiers. For *Human3* after the entire occlusion of the target, CLIP [34] is able to recover the correct track, while drift occurs when using the compared trackers, like MUSTer and LCT, which, thanks to the preserved long-term target appearance information that can help to re-detect the targets after recovery. Due to the target samples from the first frame that are used for model retraining, ELMAE [3] shows excellent results in the sequences of *Jogging* and *Suv*. Positive samples in the first frame are also used for the network update by DNCT [45], which is able to recapture the recovered targets, even if they are much smaller than normal, i.e., *Skiing*. The targets in *Lemming* and *Jogging2* simultaneously rotate and change their appearance, in the meantime both of them are in the state of occlusion. Owing to the dynamic reliability parameter that is used for occlusion detection, SRKCF [35] can more effectively handle those more complicated occlusion problems, the center location error (CLE) of which is relatively lower than its compared benchmark trackers.

(2) *Background interference*: Background clutter is also one of the most challenging factors, performances of a tracker under which situation is a key point of the measurement of its robustness. For *Basketball* and *Bolt2,* where there exist objects sharing too many characters with the true target in the surroundings, SRKCF [6,60] is able to track the true target while other compared trackers, like SRDCF, LCT, and SAMF, drift to the false ones. This is because of the fact that SRKCF has made use of the distribution of foreground and background pixels that is useful to feature update, which is combined with the parameters of the target location distance and the *PAR* to prevent similar but unrelated background pixels from contaminating the target template. MLFF [18] adopts integrated features extracted by multiple network layers to distinguish the true objects when considering that the true target is not completely the same as the false one in the background, which performs well on the challenging sequencesm such as *FaceOCC2, Football, Sylvester*, *CarDark,* and *Singer2*. Cluttered backgrounds in some frames, like frame #51 of *Davidoutdoor*, frame #146 of *Bicycle*, frame #53 of *Thusy,* and frame #105 of *Gymnastics* may impact the feature extraction of the targets therefore drifts probably appear when the sequences are tracked while using some benchmark trackers. Thanks to the approach utilized in TBE [12] that separates the target from the background and regards the background as the Gaussian model, which is able to resist many kinds of background interference, hence the appearance information of the targets can be correctly used for more concise target location. For instance, for the sequences of *Bicycle* and *Uav*, owing to the fact that initial location of the target might be incorrect because of background noises, the separated target appearance model can be used to obtain the more accurate location. For the framework of dictionary learning based ODLR [39], target dictionary and background dictionary are independently constructed while using positive and negative samples respectively describing the target and the background, which is helpful in the detection of complicated backgrounds. The tracker performs well on *Deer* sequence, while the benchmarks, such as ALSA, IVT, SCM, and VTD, do comparably poorly. The distance of the estimated target locations in two adjacent frames might be larger than normal due to the interference of the false target in the background, based on which problem, the relocation mechanism in LSHR [33] makes good use of target features that are extracted by different layers, hence it is able to accurately track some videos with background clutters i.e., *Ironman* entirely, while the benchmark trackers cannot perfectly handle the problems.

(3) *Illumination variation*: Illumination variation of a target is a kind of passive appearance change that the illumination of the target is influenced by the environment it exists in. For example, some noises, such as too light or too dark spots, caused by unusual environment illumination may appear in the target area. Due of the samples from the past latest frames used for the target dictionary update in [40], the dictionary can well encode the latest appearance of the target, especially when there are intensive changes on some features. In frame #127 of *Davidindoor,* when the tracked man suffers intense illumination variation, the proposed tracker in [40] is able to more perfectly capture his immediate appearance change as compared to SSL. Strong performances are shown by LSA [17] on the sequences of *Sylvester* and *Shaking,* in both of which there are tense

illumination changes, which can be attributed to the stable patches and valid ones that are of excellent use for the representation of deformed objects. MSRBT [54] gives more concern about the features that are more apparent for target and background discrimination, while repressing the ones not so available. It makes use of those distinguishable features for the detection of the targets in the frames with illumination variation, i.e., frame #156 of *Singer2*, frame #22 of *Crowds,* and frame #408 of *David*, while the compared benchmarks SCM, L1APG, and ALSA perform worse owing to the use of illumination-sensitive gray features. By fusing the features of color names (CN), color histograms (CH), and HoG in appropriate proportions, CLIP [34] is able to encode the appearance of a target from diverse aspects, which performs apparently better than HCF and SiamFC on *Singer2,* in which drastic illumination variation comes across, for the reason that the benchmarks have made excessive use of semantic features that are not of good use for the discrimination in that situation. Illumination-insensitive HoG feature is emphasized by TPS [14], which shows strong robustness on the frames of #528, #615, and #703 of *Sylvester*. The use of the "mean-background" that eliminates the influence of illumination variations makes TBE [12] more robust, which shows excellent results on frame #177 of *Bicycle* and frame #202 of *Woman*.

(4) *Rotation*: Rotation of a target can be regarded as a type of target appearance variation; however, other challenging factors, such as occlusion or scale variation, may occur in the meantime during target's rotation, thus in-plane and out-of-plane rotations are also hard problems to tackle with. In the sequence of *Skating*, the target athlete is rotating in-plane and out-of-plane alternatively, in addition in frame #304, it suffers intensive illumination variation; SALSC [41] is still able to capture the athlete after frame #304, while its compared benchmarks have lost the target. Excellent performance is also shown on *Car4* by SALSC. These good performances are thanks to the template update mechanism that gives new target appearances and old ones with equal importance. When considering that rotation is the appearance change of a target that its local parts are rearranged within the target area of an image, LSA [17] makes use of valid target patches for the representation of newest target appearances that have undergone in-plane or out-of-plane rotation. It tracks the targets in the sequences of *Basketball* and *Bolt,* in which out-of-plane rotation happens much more favorably correctly as compared to L1APG and Struck, which do not have the capability of rotation handling. Among the network of LSHR [33], the midst layer does the best in coping with the rotation problems, in addition features that are extracted by the shallowest layer are also adopted, thus the network is able to deal with challenging situations in mixture of low resolution and rotation, whereas excellent tracking results are performed on the sequences of *David* and *FleetFace*. In-plane rotated targets can be spontaneously separated from the surroundings, owing to the background subtraction mechanism of TBE [12]. Though new characters of the target can appear if it has undergone out-of-plane rotation, because of the principle that TBE has acquired abundant background information that is of valuable use of background discrimination, newly appeared target characters are detected as background correctly; therefore, TBE also performs much better under this situation. Robust performances are shown on frames #309 and #353 of *Polarbear* and frames of #101 and #961 of *Lemming,* whereas other benchmark trackers can hardly achieve such correct tracking.

(5) *Scale variation* and *deformation*: The scale of a target varies continuously with indeterminacy in frames, the shape of which might also vary along with its initiative scale change, because of the movement of targets and the camera. In the sequence *Sylvester*, the tracked doll severely deforms in frames #676 and #1078; DSARCF [56] can well capture the doll and precisely estimate its scale and shape, while the compared trackers lose the target or wrongly calculate the size during tracking. This is owed to the saliency information DSARCF adopts when updating the spatial weight map, after all the saliency map of a target can naturally reflect the size and shape of it. For the sequence of *Bolt* where the player deforms his body, MSRBT [54] does excellent in tracking him, owing to the local multiple feature pattern. In *Singer1* the size of the target singer constantly varies due to frequent camera distance variation between him LSA [17] tracks the singer much more correctly than the benchmarks due to the state search mechanism based on PF. However, it lacks the capability to deal with more drastic scale variations.

5.3. Analysis of Failure Cases

Target re-detection is an essential part in the whole process of tracking, into which consideration has to be put by all model update strategies. There are some failure tracks that the reappeared targets that have ever been out of the scene are retracked in vain due to the lack of re-detection mechanism by the tracker or improper update methods serving for re-detection. For instance, the walker reappears in the scene after long time of out-of-view in *Pedrstrian2*. ODLR [39] fails to capture it again due to the use of false positive samples for the dictionary construction. This is attributed to the lack of the re-detection process that ODLR has poor ability in relocating targets after heavy background drifts or target losses. Additionally, for *Suv,* where the target reappears after occlusion, MSRBT [54] does poorly in recognizing it. HCF [29] fails to recapture the targets in *Girl2* and *Lemming* when they return visible.

There remains a question of making use of valuable features of small targets in visual tracking. ODLR performs not so well on *Skiing,* owing to the fact that there is not sufficient target information for the construction of the target dictionary because of much too small size of the target, thus the target cannot be properly described by the model. For TBE [12], which puts important attention to the background, the background occupies nearly the entire image when the target is excessively small, hence appearance features of the target are hard to learn by the tracker, thus drift problems exist in some snatches of the sequences where the targets are comparably much smaller.

There are also some failure cases in some videos where situations of fast motion, rotation, background clutter, irregular illumination distribution, etc. exist. Although DSARCF [56] can handle scale variation and deformation problems perfectly due to the use of saliency information, it fails in utilizing the feature information of targets with faster moving speed, especially when the background moves together with the target, where the saliency map loses its function. For instance, the background moves upwards or downwards along with the diving athlete in *Jump*, in which the backgrounds in the adjacent frames have more differences than in the normal conditions, which prohibits the filter in the previous frame from valid detection in the current frame. The saliency map cannot work well either on the sequence of *Matrix,* where there are influences of low resolution and background clutter, bringing about target loss in tracking the later part of the sequence. For *Dragonbaby,* where the face of the baby disappears and its arm becomes distinct, the bounding box permanently drifts to the region of the arm. CLIP [34] is not to able to cope with the rotation problems in which handcrafted target features are used, leading to the drift in *MotorRolling* where the target rotates and translates rapidly simultaneously, which is also failed to track by MSRBT that also utilizes handcrafted features with limited robustness only, owing to the fact that target rotation implies the transformation of its spatial orientation. Despite the capability that PF can well calculate the states of targets in silent videos, it does not work well in estimating the states of moving objects, thus performs much poorer in the videos where targets move much more drasticly. For instance, LSA [17] wrongly estimates the target’s states in *CarScale*. Besides, LSA has poor ability in distinguishing responsible patches from occluded ones owing to the mechanism of linear regression, hence it has poor performances on the sequences with mixed challenging factors, such as *Ironman* and *MotorRolling*. Due to the fact that deeper layers of a network extract more semantic features, HCF can not discriminate the dark target singer and the bright background, since the features that are extracted by the first layer are reliable enough to complete the classification.

5.4. Module Summary

In this module, approaches of model construction and update strategies that are based on paper researches in recent years are listed and some typical performances of the trackers are briefed in the first subsection. In the second subsection, excellent performances of the recent trackers under challenging factors, which are occlusion, background interference, illumination variation, rotation, scale variation, and deformation are illustrated in order in detail. Advantages of the update strategies of each tracker are illustrated based on specific tracking cases and analyses regarding model construction and update mechanism are given to account for those merits. In the third subsection, some failure tracking cases are listed and the remaining problems of the listed trackers are explained.

Through the performance results, we can draw a conclusion that the framework of a tracker lays the foundation of its basic quality, while the update approach reflects its robustness and adaptability. Detailed conclusions are drawn by the analysis of diverse update mechanisms, which are illustrated below.

(1) Local representation of a target makes the tracker much easier to detect the local parts of the target. An independent update of each local patch guarantees that the tracking model can well capture more reliable local appearances of target local parts. On the basis of animal's selective attention mechanism, it is not necessary to fix attention on the whole object when tracking it, whereas only the typical characters of the target rather than others are sufficient for use as the attention for visual detection. In addition, it is better to design a multiple-tracker framework that each part of a target is independently tracked to reduce the complexity of training samples and the irregularity of sample distribution. Lots of researches have proved that frameworks with local patterns perform stronger robustness under many challenging situations as compared to that with mere global patterns, especially under the state of partial occlusion, although of which state the occluded parts of the target template cannot be updated, the remaining visible parts can still be used for detection and location and their corresponding parts of the template can be dynamically updated to make the model adapt to the newest target appearance. Examples of L3SCM [24], TPS [14], MSRBT [54], etc., have verified the robustness of tracking under partial occlusions. In addition, rotation problems can also be dealt with by local patterns. When in-plane rotation happens, all of the target parts remain visible and there is just the rearrangement of places order of the parts, while during out-of-plane rotation, some of target parts remain visible. Under these cases, the old visible parts contain valuable information regarding the location of target parts, thus independent update of each target part makes sure that symbolic target parts provide the most assistance for the location of the whole target. Some researches also have shown the effectiveness of local patterns in dealing with rotations, as for the instance of LSA [17], where the valid patches provide a lot of contributions to the target detection under the states of in-plane and out-of-plane rotations.

(2) The utilization of multiple features makes the tracker much more excellent in figuring out the target under some special situations. Independent update of different feature parameters can make the tracker avoid the disturbances of environmental changes, which is also an approach that disassembles the complexity of initial training samples. As different features describe a target from diverse respects, the contribution of each feature is not constant in different periods of tracking [58]; hence, it is better to adopt feature-specific update methods. As for the instance of illumination variation, the target passively changes its appearance along with the illumination change of the environment, during which period some features have undergone apparent changes, i.e., gray feature [54], while some do not change so much, i.e., HoG [14]. In this case, illumination-insensitive features, like HoG, are of better use for target detection and larger weights should be given on it, while features sensitive to illumination, such as gray feature, should be given smaller weights to reduce the impact of noise.

(3) Features that are extracted by layers of different depths in a deep neural network also do different performances on tracking, and the highest-level features are not always the most effective. In the process of visual tracking, the only work is to separate the target from the background and

then locate it, rather than obtaining the semantic features of the target and its surroundings, thus sometimes features that are extracted by deeper layers are less important than shallower ones. The failure case of tracking the actor in *Singer2* by HCF [29] has indicated the drawback of the high-level feature extracted by deep layers, whereas the features that are extracted by the first layer do the best performance on the contrary under this situation. The example of LSHR [33] has also explained that each layer has its own excellence in the discrimination, where features that are extracted by the first layer are best at dealing with low resolution problems, while features by the third layer do the best in handling with rotations and features by the fifth (deepest) layer performs best in occlusion cases. The update strategy that takes the contribution of each layer into consideration and gives full play to the advantages of each layer improves the robustness of the tracking network to a great extent.

(4) Target re-detection is an especially essential part in tracking, and the positive samples from the first frame reused for update help with recovery of the target. Tracking failures of MSRBT [54], HCF [29], and ODLR [39] have explained this significance. The positive and negative samples are used for the dictionary construction in [51], in which the dictionaries are adopted for the target re-detection. The RDLT tracker does perfectly under the situations of out-of-view and full occlusion thanks to the re-detection mechanism and update policy. Due to the fact that a target might also vary its appearance in the process of temporary leaving off from the scene or being fully occluded, it is not responsible enough to merely use the latest appearance models in the moment before its disappearance. Like the training approach in object detection, theoretically target images, including all of the target appearances ever appeared, should be used as the retraining sample set. However, though positive samples from some best tracked frames can also help with re-detection, target samples from the first frame are believed as the most credible and share some characters with the recovered target, even if the appearance of them may not be so close due to the impact of noise and other environmental disturbances, thus the update methods that take the positive samples from the first frame into account make better performances on target re-detection. Instances of ELMAE [29] and DNT [27] have also verified this effectiveness.

(5) An excellent update strategy of the background models helps with more precise target location. Owing to the continuity of target movement in the background, positions of the target in the surroundings in adjacent frames are very close; hence, characters of the surroundings of the target contain valuable information for target location in the next frame. Examples of BoW (BoF) based models are supportive of this conclusion. It is good to adopt dense sampling to make each surrounding region more representative since the background has more diversity appearances.

(6) The saliency feature of the target provides the tool for the estimation of target scale and shape. The scale and shape of the target in the saliency map highly reflects those in the original image due to the characteristic of the target that it should be a salient object, thus the saliency feature does well in handling scale variation and deformation. The precise estimation of the target scale and shape by DSARCF [56] shows its function.

6. Summary and Outlooks

This review has given detailed analysis of the visual tracking model update approaches in recent years, where discussions about target model update occasions and strategies as well as approaches of background model update are illustrated in order, and specific performances of sequence tracking are then exampled. The merits and drawbacks of the listed trackers in recent researches are illustrated afterwards and conclusions regarding the performances with respect to model construction and update are briefed. In light of the problems remaining in the latest tracking performances posing challenges to future researches, to make future tracker frameworks more applicable, focuses with respect to the model update training of visual tracking should be fixed on the following aspects:

(1) Adoption of the background information should be further enhanced and algorithms for dynamic background appearance model update need to be designed. In view of the truth that a target must exist in a specific environment, information regarding the background that surrounds the target provides sufficient information for target location, which can wonderfully help with

discriminating the true target and the similar objects in the surroundings. Encoding and updating the background model should well be respectively conceived from the angles of the global pattern and local ones, of which the former gives the requirement that relationships among each background parts should be encoded for the holistic description, which provides useful information for the rough location of the target. When the tracked target becomes much too smaller than that in the first frame, the holistic character of the surrounding background rather than the target itself is better to be tracked, hence the problem of the model construction of small targets can be greatly alleviated. Besides, the hard problem of rapidly moving background can also be well settled. Therefore, a network that has the ability to recognize the position and distance relationships among different background parts should better be designed, which should be an application in object position recognition. The latter namely local patterns requires that the set of characters in the regions of surroundings which includes the most evident symbols for target location should be searched out as the auxiliary feature, which is close to the mechanism of animal's selective visual attention that symbolic background areas ought to be given more visual attentions, thus the moving speed of a target can be well estimated with the help of these background auxiliaries. Based on the tool of the target response map, the response map of background regions should be made use of to decide whether the holistic model or the local model needs to be updated.

(2) Saliency information should be adopted as an important feature. As random variations of the target scale and shape also constitute the challenging factors, though some target state estimation models, like PF, scale CF, etc., perform well in calculating the size and shape of targets, they are not always credible due to the extent of target movement and restriction of datum complexity, whereas the saliency map of the target naturally represents the shape and size of it in the original image; therefore, it can be regarded as a responsible feature for the state estimation of the target, which gives the guidance of target detection under partial occlusions that lays the foundation of target appearance update in this situation. Although there are failures that the saliency map does not work well on some sequences, it is better to be used after the raw detection of the target for further state estimation instead of using as an appearance template before target search, and it should be preserved as a state template for the reference of target state calculation in the next frame.

(3) It makes a tracker more robust to make adequate adoption of features extracted by different layers to achieve more responsible target detection and location, and a multiple-feature based template model should be utilized. Like handcrafted features, neural network features that are extracted by different layers express the target and background from different aspects, making different contributions to the tracking performance, hence the tracking performances by different layers ought to be considered for update. To make deep neural network frameworks give more adequate play in many computer vision applications, it ought to be an excellent idea to make use of layers of any depth for specific usages. In light of the fact that the feature that was extracted by one deeper layer is further processed and extracted from that by the previous shallower layer, the complexity of the feature information increases with the depth of the layer. Yet, the feature information by the higher level sometimes provides more contribution, while in other cases lower level features contribute the most on the contrary. Inspired by the cascade template update approach in [61], cascade adoption of the features by different depth of the layers should be taken into consideration to assess the tracking performance using the performance score of each layer in depth order. Meanwhile, the template corresponding to each layer should be set up. If the performance score of the shallower layer is higher than a predefined threshold, the template corresponding to this layer needs to be updated meanwhile update of the templates corresponding to deeper layers should be temporarily disabled. Target matching should also be done in the cascade mode. if the template corresponding to a layer matches to the target candidates with too high confidence, those to the matching of the templates corresponding to the higher levels should also be stopped in this turn. The proposed cascade method is able to prohibit the interference from irrelevant features, thus time expense aroused from feature selection can be reduced.

Author Contributions: Conceptualization, D.Y.W. and W.C.; methodology, W.C. and W.D.F.; validation, W.D.F. and T.F.S.; formal analysis, D.Y.W. and W.C.; investigation, D.Y.W. and T.J.C.; resources, W.C.;

writing—original draft preparation, D.Y.W.; writing—review and editing, D.Y.W. and W.D.F.; visualization, D.Y.W.; supervision, W.C.; project administration, W.C.; funding acquisition, W.C.

Funding: This word was funded by National Natural Science Foundation of China, grant number 51874300, National Natural Science Foundation of China and Shanxi Provincial People's Government Jointly Funded Project of China for Coal Base and Low Carbon, grant number U1510115, and the Open Research Fund of Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, grant numbers 20190902 and20190913.The APC was funded by51874300, U1510115 and 20190902.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, X.; Meng, L. A Survey of Object Tracking Algorithms. *Acta Autom. Sin.* **2019**, *20*, 1–15.
2. Zhu, G.; Porikli, F.; Li, H. Not All Negatives Are Equal: Learning to Track With Multiple Background Clusters. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 314–326.
3. Han, Y.; Deng, C.; Zhao, B. High-Performance Visual Tracking with Extreme Learning Machine Framework. *IEEE Trans. Cybern.* **2018**, 1–12.
4. Dong, X.; Shen, J.; Yu, D.; Wang, W.; Liu, J.; Huang, H. Occlusion-Aware Real-Time Object Tracking. *IEEE Trans. Multimed.* **2017**, *19*, 763–771.
5. Ilic, S.; Holzer, S.; Navab, N.; Tan, D.; Pollefeys, M. Efficient Learning of Linear Predictors for Template Tracking. *Int. J. Comput. Vis.* **2015**, *111*, 12–28.
6. Ilic, S.; Holzer, S.; Navab, N. Multipayer Adaptive Linear Predictors for Real-Time Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 105–117.
7. Del Bimbo, A.; Pernici, F. Object Tracking by Oversampling Local Features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2538–2551.
8. Ren, Y.; Huang, R. Visual Tracking Using Spatio-Temporal Context Template Set Learning. In Proceedings of the 2017 9th IEEE International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 6–8 May 2017; pp. 1496–1500.
9. Qiao, Y.; Niu, X. Context-Based Occlusion Detection for Robust Visual Tracking. Internation Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3655–3658.
10. Gu, Y.; Niu, X.; Qiao, Y. Robust Visual Tracking via Adaptive Occlusion Detection. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 2242–2246.
11. Niu, X.; Fang, X.; Qiao, Y. Robust visual tracking via occlusion detection based on staple algorithm. In Proceedings of the 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, Australia, 17–20 December 2017; pp. 1051–1056.
12. Zhang, S.; Zhao, S.; Zhang, L. Towards Occlusion Handling: Object Tracking with Background Estimation. *IEEE Trans. Cybern* **2018**, *48*, 2086–2099.
13. Wang, H.; Zhang, X.; Yu, L.; Wang, X. Research on Mean Shift Tracking Algorithm Based on Significant Features and Template Updates. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018; pp. 1199–1203.
14. Huang, L.; Shao, L.; Ma, B.; Shen, J.; He, H.; Porikli, F. Visual Tracking by Sampling in Part Space. *IEEE Trans. Image Process.* **2017**, *26*, 5800–5810.
15. Lauer, M.; Tian, W. Tracking Objects with Severe Occlusion by Adaptive Part Filter Modeling--in Traffic Scenes and Beyond. *IEEE Intell. Trans. Syst. Mag.* **2018**, *10*, 60–73.
16. Vipin Krishnam, C.V.; Ramya, K.V. Object Tracking Via Boosted Cascade of Simple Features and Coarse and Fine Structural Local Sparse Appearance Models. In Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 3–5 April 2018; pp. 693–697.
17. Nai, K.; Li, Z.; Li, G.; Wang, S. Robust Object Tracking via Local Sparse Appearance Model. *IEEE Trans. Image Process.* **2018**, *27*, 4958–4970.
18. Kuai, Y.; Wen, G.; Li, D. Learning Fully Convolutional Network for Visual Tracking With Multi-Layer Feature Fusion. *IEEE Access* **2019**, *7*, 25915–25923.

19. Zou, Q.; Lin, S.; Du, Y. High Confidence Updating Strategy on Staple Trackers. In Proceedings of the 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, China, 24–27 August 2018; pp. 238–241.
20. Soldic, M.; Marcetic, D.; Maracic, M.; Mihalic, D.; Ribaric, S. Real-time face tracking under long-term full occlusions. In Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis, Ljubljana, Slovenia, 18–20 September 2017; Volume 1, pp. 147–152.
21. Zhu, Y.; Wen, J.; Zhang, L.; Wang, Y. Visual Tracking with Dynamic Model Update and Results Fusion. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2685–2689.
22. Li, Z.; Zhang, J.; Zhang, K.; Li, Z. Visual Tracking With Weighted Adaptive Local Sparse Appearance Model via Spatio-Temporal Context Learning. *IEEE Trans. Image Process.* **2018**, *27*, 4478–4489.
23. Qiu, S.; Zhang, J.; Qing, S.; Dong, J.; Guo, W. Object Tracking Method Based on Semi Supervised Extreme Learning. In Proceedings of the 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE), Changchun, China, 6–8 July 2018; pp. 308–312.
24. Elharrouss, O.; Moujahid, D.; Tairi, H. Visual Object Tracking Via the Local Soft Cosine Similarity. *Pattern Recognit. Lett.* **2018**, *110*, 79–85.
25. Yuefang Gao ZH, Henry W. F., Yuying Zhong, Xuhong TIan, Liang Lin. Unifying Temporal Context and Multi-feature with Update-pacing Framework for Visual Tracking. IEEE Transactions on Circuits and Systems for Video Technology. 2019.
26. Ma, D.; Bu, W.; Xie, Y.; Cui, Y.; Wu, X. Segmentation-Guided Tracking with Prior Map Decision. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2014–2019.
27. Chi, Z.; Li, H.; Lu, H.; Yang, M.-H. Dual Deep Network for Visual Tracking. *IEEE Trans. Image Process.* **2017**, *26*, 2005–2015.
28. Han, B.; Nam, H. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 4293–4302.
29. Ma, C.; Huang, J.-B.; Yang, X.; Yang, M.-H. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
30. Alatan, A.A.; Gundogdu, E. Good Features to Correlate for Visual Tracking. *IEEE Trans. Image Process.* **2018**, *27*, 2526–2540.
31. Dai, K.; Wang, Y.; Yan, X.; Huo, Y. Fusion of Template Matching and Foreground Detection for Robust Visual Tracking. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2720–2724.
32. Luo, L.; Huang, D.; Chen, Z.; Wen, M.; Zhang, C. Applying Detection Proposals to Visual Tracking for Scale and Aspect Ratio Adaptability. *Int. J. Comput. Vis.* **2017**, *122*, 524–541.
33. Tang, F.; Lu, X.; Zhang, X.; Luo, L.; Hu, S.; Zhang, H. Adaptive convolutional layer selection based on historical retrospect for visual tracking. *IET Comput. Vis.* **2019**, *13*, 345–353.
34. Hu, Q.; Liu, H.; Li, B.; Guo, Y. Robust Long-Term Tracking Via Instance Specific Proposals. *IEEE Trans. Instrum. Meas.* **2018**, *20*, 1–13.
35. Hu, G.; Liu, Q.; Islam, M.M. Robust Visual Tracking with Spatial Regularization Kernelized Correlation Filter Constrained by a Learning Spatial Reliability Map. *IEEE Access* **2019**, *7*, 27339–37351.
36. Guo, J.; Liu, J.; Hi, S. Correlation Filter Tracking Based on Adaptive Learning Rate and Location Refiner. *Opt. Prec. Eng.* **2018**, *26*, 2100–2111.
37. Zhang, K.; Wang, W.; Lv, M. Robust Visual Tracking Based on Adaptive Extraction and Enhancement of Correlation Filter. *IEEE Access* **2019**, *7*, 3534–3546.
38. Qin, X.; Yang, M.-H.; Wang, G.; Zhong, F.; Liu, Y.; Li, H.; Peng, Q. Visual Tracking via Sparse and Local Linear Coding. *IEEE Trans. Image Process.* **2015**, *24*, 3796–3809.
39. Zhou, T.; Liu, F.; Bhaskar, H.; Yang, J. Robust Visual Tracking via Online Discriminative and Low-Rank Dictionary Learning. *IEEE Trans. Cybern.* **2018**, *48*, 2643–2655.
40. Lu, X.; Yi, S.; He, Z.; Wang, H.; Chen, W.-S. A New Template Update Scheme for Visual Tracking. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016; pp. 243–247.

41. Qi, Y.; Qin, L.; Zhang, J.; Zhang, S.; Huang, Q.; Yang, M.-H. Structure-Aware Local Sparse Coding for Visual Tracking. *IEEE Trans. Image Process.* **2018**, *27*, 3857–3869.
42. Gong, C.; Liu, F.; Zhou, T.; Fu, K.; He, X.; Yang, J. Visual Tracking Via Nonnegative Multiple Coding. *IEEE Trans. Multimed.* **2017**, *19*, 2680–2691.
43. Zeng, F.; Ji, Y.; Levine, M.D. Contextual Bag-of-Words for Robust Visual Tracking. *IEEE Trans. Image Process.* **2018**, *27*, 1433–1447.
44. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Zajc, L.C.; Vojir, T.; Hager, G.; Lukezic, A.; Eldesokey, A.; et al. The Visual Object Tracking VOT2017 Challenge Results. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 1949–1972.
45. Huo, H.; Lu, X.; Fang, T.; Zhang, H. Learning Deconvolutional Network for Object Tracking. *IEEE Access* **2018**, *6*, 18032–18041.
46. Wang, L.; Zhang, L.; Wang, J.; Yi, Z. Memory Mechanisms for Discriminative Visual Tracking Algorithms with Deep Neural Networks. *IEEE Trans. Cogn. Dev. Syst.* **2019**, 1.
47. Zhang, S.; Lan, X.; Yao, H.; Zhou, H.; Tao, D.; Li, X. A Biologically Inspired Appearance Model for Robust Visual Tracking. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *28*, 2357–2370.
48. Ge, D.; Song, J.; Qi, Y.; Wang, C.; Miao, Q. Self-Paced Dense Connectivity Learning for Visual Tracking. *IEEE Access* **2019**, *7*, 37181–37191.
49. Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.-H. Robust Visual Tracking via Convolutional Networks without Training. *IEEE Trans. Image Process.* **2016**, *25*, 1.
50. Deng, C.; Wang, B.; Lin, W.; Huang, G.-B.; Zhao, B. Effective visual tracking by pairwise metric learning. *Neurocomputing* **2017**, *261*, 266–275.
51. Zhou, W.; Wang, N.; Li, H. Reliable Re-Detection for Long-Term Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 730–743.
52. Sui, Y.; Wang, G.; Zhang, L.; Yang, M.-H. Exploiting Spatial-Temporal Locality of Tracking via Structured Dictionary Learning. *IEEE Trans. Image Process.* **2018**, *27*, 1282–1296.
53. Jin, X.; Zhang, J.; Sun, J.; Wang, J.; Li, K. Dual Model Learning Combined with Multiple Feature Selection for Accurate Visual Tracking. *IEEE Access* **2017**, *20*, 1–9.
54. Zhang, S.; Lan, X.; Yuen, P.C.; Chellappa, R. Learning Common and Feature-Specific Patterns: A Novel Multiple-Sparse-Representation-Based Tracker. *IEEE Trans. Image Process.* **2018**, *27*, 2022–2037.
55. Han R.; Guo, Q.; Feng, W. Content-Related Spatial Regularization for Visual Object Tracking. In proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018.
56. Feng, W.; Han, R.; Guo, Q.; Zhu, J.; Wang, S. Dynamic Saliency-Aware Regularization for Correlation Filter-Based Object Tracking. *IEEE Trans. Image Process.* **2019**, *28*, 3232–3245.
57. Hou, Z.; Wang, X.; Yu, W.; Jin, Z.; Zha, Y.; Qin, X. Online Scale Adaptive Visual Tracking Based on Multilayer Convolutional Features. *IEEE Trans. Cybern.* **2019**, *49*, 146–157.
58. Zhang, S.; Qi, Y.; Qin, L.; Huang, Q.; Yao, H.; Lim, J.; Yang, M.-H. Hedging Deep Features for Visual Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1116–1129.
59. Feng, D.; Gao, F.; Wang, X.; Wang, G.; Dai, H. Cascaded Iterative Training Model and Parallel Multi-Classifiers for Visual Object Tracking. *IEEE Access* **2019**, *7*, 63099–63112.
60. Wang, T.; Ling, H.; Lang, C.; Feng, S.; Jin, Y.; Li, Y. Constrained Confidence Matching for Planar Object Tracking. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 659–666.
61. Wang, Y.; Dai, K.; Yan, X. Long-Term Object Tracking Based on Siamese Network. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3640–3643.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).