Journal of Management & Entrepreneurship
ISSN 2229-5348

UGC Care Group I Journal
Vol-09 Issue-02 Nov 2020

Complementary Features in a Hybrid Cascade Filter for Visual Tracking

Snehasis Dey
Dept. of Electronics & Telecommunication Engineering,
College of Engineering Bhubaneswar

***Abstract*— Aspects of the thing are described by their many features. To get good performance, the right features must be specifically tailored for each visual tracking application. In this letter, we suggest combining deep and handmade features into a hybrid cascade filter to maximize their respective advantages. In order to improve localization accuracy and robustness, we combine the deep representation with manually created features and construct a hybrid cascade structure with several observation models. To further reduce the computational cost, a coarse-to-fine searching approach is employed. Comprehensive test outcomes on a pair of reference datasets demonstrate that the suggested approach outperforms the most advanced trackers.**

***Index Terms*—Deep features, hybrid cascade filter, handcrafted features, visual tracking.**

I. Introduction

VISUAL tracking is a basic research topic in the field of computer vision and has numerous applications such as video surveillance, visual guidance, and human-computer interaction. It is mainly focused on estimating the states of an object in a video, given only its initial bounding box. This task is difficult, primarily because that the training data available for learning the object appearance model online is limited. Existing methods rely on rich feature representations to address this fundamental challenge [1], [2]. Consequently, feature extraction plays an important role in a tracker. Using proper features can dramatically improve the tracking performance.

At present, the features used in visual tracking are divided into two categories: handcrafted features and deep features. Handcrafted features have long been employed for tracking task [3], [4], [5], which mainly include gray level, Color Names (CN) and Histogram of Oriented Gradients (HOG), etc. They are easy to calculate and explain, and contain rich low-level visual information. Recently, the focus has shifted to more powerful deep features, such as CNN features [6]–[8]. CNN features consist of the outputs from all different convolutional layers. The outputs in the earlier layers (earlier CNN features) retain fine-grained spatial details, while the ones in the latter layers (latter CNN features) encode high-level semantic information which is invariant to complex appearance changes and clutter. However, their disadvantage is also obvious. CNN generally trades spatial resolution and computational cost for increased high-level invariance to appearance changes, hampering accurate localization. Therefore, many trackers complement the deep representation with shallow features have been proposed [6]–[11].

Considering that the outputs of convolutional layers in different levels can complement each other, Ma *et al.* [6], Li *et al.* [7] and Danelljan *et al.* [8] made full use of features from different layers through a variety of fusion strategies to achieve precise localization, which was a further step in understanding of the layered architecture of deep network. Zhang *et al.* [9] first presented the multi-task correlation filter (MCF) that exploited interdependencies among different CNN features to learn correlation filters jointly, and then combined it with a particle filter to effectively overcome large-scale variation. Sun *et al.* proposed the DRT method [10] to jointly model the discrimination and reliability information using handcrafted and deep features. In [11], Dai *et al.* applied two kinds of correlation filter models. One exploited ensembles of handcrafted and deep features to determine the optimal position. The other worked on multi-scale handcrafted features to estimate the optimal scale.

It can be found that for some targets, e.g., small-size ones, even earlier CNN features could not capture the useful spatial information while handcrafted features could. In fact, trackers using handcrafted features can still provide competitive results, even better than many deep trackers on standard benchmarks [2]. Furthermore, many trackers which complement the deep representation with low-level activations or handcrafted features are focused on improving localization accuracy, but their robustness are not ideal. This raises the question of how to optimally fuse handcrafted features and deep features to achieve both accuracy and robustness.

From the above analysis, we propose a hybrid cascade filter (HCF) with complementary features for visual tracking. The main points are as follows:

1) propose an HCF to fuse handcrafted features and latter CNN features using multiple observation models (observers) to achieve better robustness;
2) use a coarse-to-fine searching strategy to effectively reduce the computational cost.

Extensive experimental results demonstrate that the proposed method performs better on two benchmark datasets.

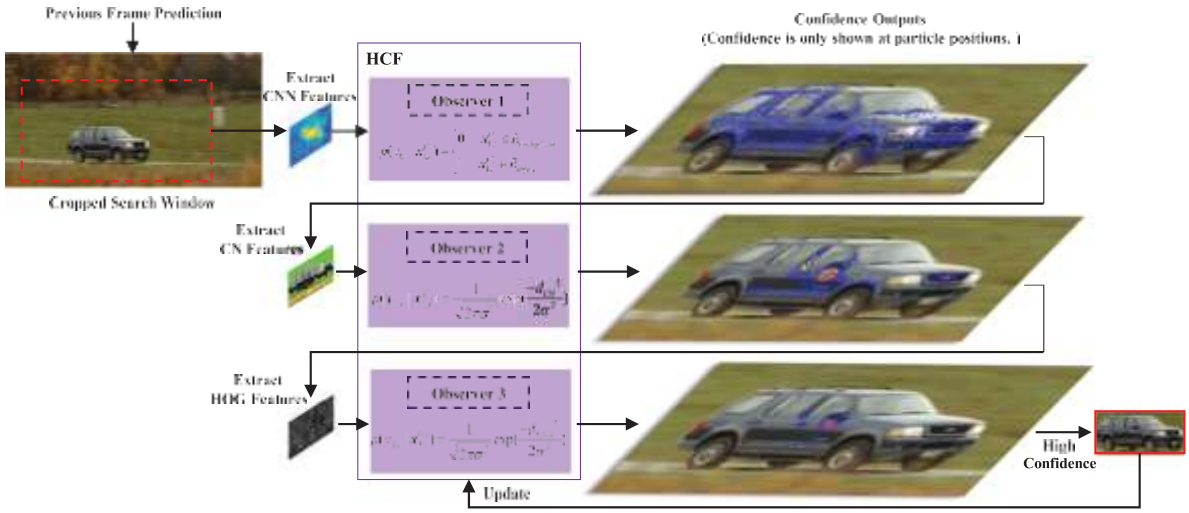

Fig. 1. Pipeline of the proposed algorithm. Red dots denote the particles with high confidence, while blue ones denote low confidence.

II. Proposed Algorithm

Cascade filter builds an efficient cascade structure using multiple observers to obtain a more robust model for visual tracking. It is also due to such structure that different features can be coupled tightly. However, tracking through the conventional cascade structure is like exhaustively searching for uniformly distributed particles over the whole state space. It can be found that cascade particle filter [12] has solved this problem, but it is not suitable for visual tracking because of its fine search at each stage. Consequently, in order to fuse complementary features, we propose the HCF which combines the merits of cascade filter and cascade particle filter.

An overview (Fig. 1) of the proposed algorithm is as follows:

1) For a given image, we crop the search window centered at the estimated position from previous frame.
2) HCF has 3-stage cascade structure, and at each stage we use proper features to establish discriminative observer. Observer 1 that captures the category information of the target for rough localization is established based on latter CNN features. Observer 2 that focuses on correcting the results from previous stage is established based on CN features. Observer 3 that possesses more spatial details for accurate localization is established based on HOG features.
3) Three confidence outputs are generated by Observer 1, Observer 2 and Observer 3, respectively. Each observer will make more accurate judgment based on the confidence output from previous stage to improve robustness.
4) We search the multi-stage confidence outputs to infer the target in a coarse-to-fine manner.
5) The current frame prediction is used to update Observer 2 and Observer 3 respectively.

A. HCF

The proposed HCF is based on Bayesian sequential importance sampling, where a finite set of weighted particles is used to recursively approximate the posterior distribution of state variables. Let $\mathbf{x}_t$ and $\mathbf{y}_t = (y_{1,t}, \ldots, y_{m,t})$ denote the state variable of the object at time t and its observations respectively.

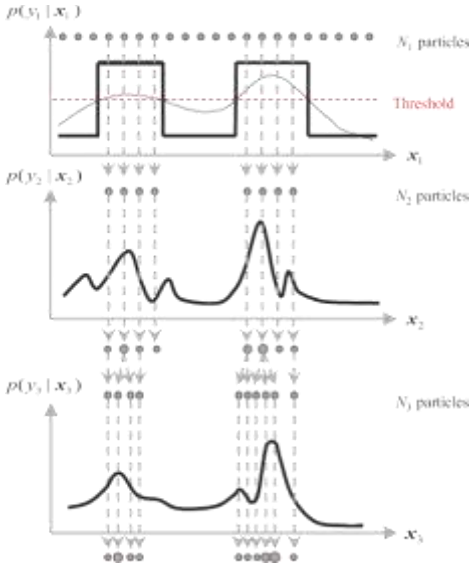

Fig. 2. Illustration of HCF.

Assuming observations are conditionally independent, we have

$$p(\mathbf{y}_t|\mathbf{x}_t) = p(y_{1,t}, \ldots, y_{m,t}|\mathbf{x}_t) = \prod_{k=1}^{m} p(y_{k,t}|\mathbf{x}_t). \quad (1)$$

Here, multiple observers $p(y_{k,t}|\mathbf{x}_t)$ are cascaded. HCF can update the particle weight by $p(y_{k,t}|\mathbf{x}_t)$ at each stage.

As shown in Fig. 2, HCF has 3 stages, and each stage includes two steps: prediction and update. At the kth stage, we have already obtained a weighted particle set $P_{k-1,t} = \{\mathbf{x}_{k-1,t}^{(i)}, \omega_{k-1,t}^{(i)}\}_{i=1}^{N_{k-1}}$ from the (k-1)th stage, where N_{k-1} is the number of particles.

Prediction: when $k = 2$, directly obtain $\{\mathbf{x}_{k,t}^{(i)}, 1/N_k\}_{i=1}^{N_k}$, where $\mathbf{x}_{k,t}^{(i)} \leftarrow \mathbf{x}_{k-1,t}^{(i)}$, s.t. $\omega_{k-1,t}^{(i)} = 0$; when $k = 3$, take the posterior distribution from previous stage as the proposal distribution sampling particles, that is, resample the particle set $P_{k-1,t}$ to obtain $\{\mathbf{x}_{k,t}^{(i)}, 1/N_k\}^{N_k}$. In order to avoid the loss of diversity

of particle set, Gaussian diffusion is added in the resampling stage.

Update: when observation $y_{k,t}$ is available, the weight of $\mathbf{x}_{k,t}^{(i)}$ could be updated as

$$\omega_{k,t}^{(i)} = p(y_{k,t}|\mathbf{x}_{k,t}^{(i)}), \tag{2}$$

when $k = 1$, define $R_{background}$ and R_{object} as the regions of background and object,

$$p(y_{1,t}|\mathbf{x}_{1,t}^{(i)}) = \begin{matrix} 0 & \mathbf{x}_{1,t}^{(i)} \in R_{background} \\ 1 & \mathbf{x}_{1,t}^{(i)} \in R_{object} \end{matrix}. \tag{3}$$

Through the above two steps, the particle set $P_{k,t} = \{\mathbf{x}_{k,t}^{(i)}, \omega_{k,t}^{(i)}\}_{i=1}^{N_k}$ is obtained. Such process is repeated for 3 stages to obtain the posterior probability $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \sim \{\mathbf{x}_{3,t}^{(i)}, \omega_{3,t}^{(i)}\}_{i=1}^{N_3}$

In conclusion, the HCF proposed in this letter has two advantages:

1) An efficient structure, which is built by multiple cascaded observers, improves the robustness of the tracker and fuses different features for better accuracy.
2) A coarse-to-fine searching strategy is adopted combining the merits of cascade filter and cascade particle filter.

B. Observers

The observer used to measure the similarity between the particles and targets plays an important role in visual tracking. In this letter, we exploit latter CNN features and handcrafted features to establish the 3-stage observers respectively.

1) Observer 1: Latter CNN features containing the semantic information have significant advantages in inter-class classification. A CNN model [13] is used as the classifier, which focuses on separating the target from the background and roughly localizing the target. The output of the CNN classifier is a binary map. Each output pixel represents the category of the corresponding input region. A pixel whose corresponding input belongs to the target is set to 1, otherwise 0. The offline pre-training and online fine-tuning of the CNN classifier are referred to [13]. Since it is used for coarse searching, update is not necessary. Observer 1 is modeled as a binary function based on the classifier's output as shown in equation (3). The CNN classifier is effective as the observer of coarse stage, which can fast eliminate most of the invalid candidates to achieve the purpose of speeding up.

2) Observer 2: CN features are computationally efficient, and possess a certain amount of photometric invariance while maintaining high discriminative power. We use CN features to model Observer 2 similar to [14], which focuses on correcting the results of coarse localization from previous stage. For any particle, the similarity of CN templates between the target and the particle is measured by Bhattacharyya Distance d_{CN} [14]. Observer 2 is modeled as

$$p(y_{2,t}|\mathbf{x}_{2,t}^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{d_{CN}^2}{2\sigma^2}\right), \tag{4}$$

where σ^2 is the scaling factor. Finally, Observer 2 should be updated, that is, the target template is replaced by a weighted combination of the current frame prediction and the old template.

3) Observer 3: HOG features capture edge or gradient structural information that is characteristic of local shape. We use HOG features to model Observer 3 for accurate localization. For any particle, the similarity of HOG templates between the target and the particle is measured by Bhattacharyya Distance d_{HOG}. Observer 3 is shown in equation (4), where d_{CN} is replaced by d_{HOG}. The update of Observer 3 is similar to that of Observer 2.

TABLE I
MODEL ANALYSIS

Dataset		HCF-HC	HCF-VGGNet	HCF-ResNet	StdCF	Ours
AUC	OTB-50	59.2	67.3	68.6	60.7	**69.9**
	OTB-100	55.8	65.0	65.9	57.9	**67.8**

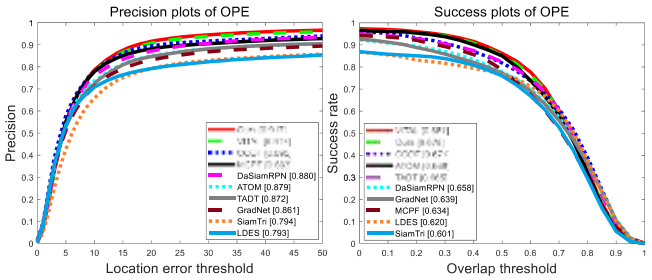

Fig. 3. Precision and success plots.

C. HCF Tracker

Our method uses HCF as the tracking framework to effectively combine latter CNN features, CN features and HOG features. We present main steps of the proposed algorithm in Algorithm 1.

Algorithm 1: Visual Tracking Based on HCF.

1: **Input:** At time $t-1$, the tracking result is $\mathbf{x}_{t-1}$.
2: **Prediction:** At time t, crop the search region using $\mathbf{x}_{t-1}$ as the center, and simulate $\mathbf{x}_{1,t}^{(i)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1})$, $i = 1, 2, \ldots, N_1$, where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the state transition distribution. Note that $\mathbf{x}_{1,t}^{(i)}$ should be diffused enough to cover the current search region.
3: **Weights update at stage 1:** For $i = 1, 2, \ldots, N_1$, calculate $\omega_{1,t}^{(i)} = p(y_{1,t}|\mathbf{x}_{1,t}^{(i)})$, and obtain the particle set $\{\mathbf{x}_{1,t}^{(i)}, \omega_{1,t}^{(i)}\}_{i=1}^{N_1}$.
4: **Coarse search:** remove the particles with $\omega_{1,t}^{(i)} = 0$ to obtain the particle set $\{\mathbf{x}_{2,t}^{(i)}, 1/N_2\}_{i=1}^{N_2}$. Note that the particle number is reduced greatly from N_1 to N_2.
5: **Weights update at stage 2:** For $i = 1, 2, \ldots, N_2$, calculate $\omega_{2,t}^{(i)} = p(y_{2,t}|\mathbf{x}_{2,t}^{(i)})$, and obtain the particle set $\{\mathbf{x}_{2,t}^{(i)}, \omega_{2,t}^{(i)}\}_{i=1}^{N_2}$.
6: **Resample + Gaussian diffusion:** simulate $\alpha_j \sim \{\omega_{2,t}^{(i)}\}_{i=1}^{N_2}$ and replace $\{\mathbf{x}_{2,t}^{(i)}, \omega_{2,t}^{(i)}\}_{i=1}^{N_2}$ with $\{\mathbf{x}_{2,t}^{(\alpha_j)}, 1/N_3\}_{j=1}^{N_3}$; simulate $\mathbf{x}_{3,t}^{(i)} \sim g(\mathbf{x}_{3,t}|\mathbf{x}_{2,t}^{(i)})$, $i = 1, 2, \ldots, N_3$, where g is a 0-mean Gaussian. Note that the particle number is reduced from N_2 to N_3.
7: **Weights update at stage 3:** For $i = 1, 2, \ldots, N_3$, calculate $\omega_{3,t}^{(i)} = \frac{p(y_{3,t}|\mathbf{x}_{3,t}^{(i)})}{g(\mathbf{x}_{3,t}^{(i)}|\mathbf{x}_{2,t}^{(i)})}$.
8: In the particle set $\{\mathbf{x}_{3,t}^{(i)}, \omega_{3,t}^{(i)}\}_{i=1}^{N_3}$, select the particle with maximum weight as the final tracking result.
9: **Output:** the tracking result $\mathbf{x}_t$.

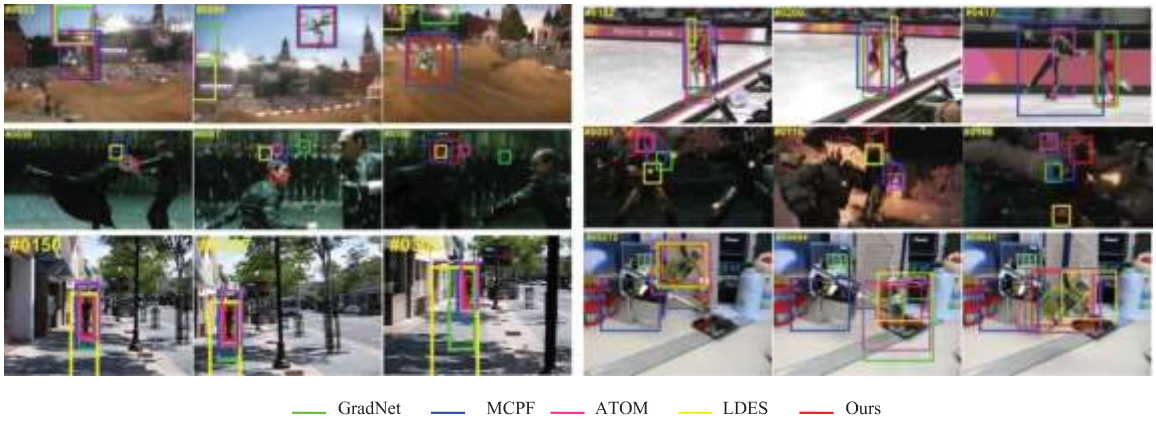

Fig. 4. Qualitative evaluation on 6 challenging sequences (from left to right and top to down are *MotorRolling*, *Skating2-1*, *Matrix*, *Ironman*, *Human9* and *Board*, respectively).

III. Experimental Results

Our tracker is implemented in MATLAB using MaConvNet on a computer with an Intel (R) core (TM) 3.19 GHz CPU and a GeForce GTX Titan X GPU. The numbers of particles adopted in each stage are 3000, 400, and 100, respectively. We evaluate our method on two benchmark datasets: OTB-50 [15] and OTB-100 [16]. For these benchmarks, we employ the one-pass evaluation (OPE) and two metrics: precision and success plots. Area under curve (AUC) and precision score at 20 pixels threshold (PS) are used as quantitative analysis indicators.

A. Model Analysis

In our tracker, we propose an HCF to exploit interdependencies among different features (CNN, CN and HOG). With the same experimental settings, we have five different trackers including HCF-HC, HCF-VGGNet, HCF-ResNet, StdCF and Ours. Here, HCF-HC is HCF using the handcrafted features (CN, HOG and Haar-like), HCF-VGGNet is HCF using the features extracted on convolutional layers conv3-4, conv4-4, and conv5-4 of the VGGNet, HCF-ResNet is HCF using the convolutional feature maps from the second, third and fourth convolutional blocks in ResNet-50, and StdCF is our tracker using standard cascade filter instead of HCF.

Table I shows that both our selected features and HCF can improve the tracking performance. First, combining latter CNN features with handcrafted features can obtain better performance. Compared with HCF-HC, our tracker achieves much better performance with about 10.7% and 12% improvement with AUC metric on the OTB-50 and OTB-100 datasets. Compared with HCF-VGGNet / HCF-ResNet, which both use CNN features extracted from multiple layers, our tracker achieves about 2.6%/1.3% and 2.8%/1.9% improvement on the OTB-50 and OTB-100 datasets. Second, HCF can improve the tracking performance. Compared with StdCF, which has the same observers as ours, our tracker achieves about 9.2% and 9.9% improvement on the OTB-50 and OTB-100 datasets.

B. Quantitative Evaluation

We compare our method with 9 state-of-the-art trackers including TADT [7], ATOM [17], GradNet [18], LDES [19], VITAL [20], DaSiamRPN [21], Siamtri [22], MCPF [9] and CCOT [8]. Fig. 3 shows precision and success plots over all 100 sequences on the OTB-100 dataset using OPE. In the legend, we report the AUC score and PS for each tracker. In the precision and success plots, the proposed tracker achieves the AUC score of 67.8% and PS of 91.7%. Among the compared trackers, our method shows comparable results to VITAL, and compared with CCOT, the performance gain is 0.7% and 2.2% in terms of AUC and PS, respectively. In Table II, we summarize the speeds of top-ranked trackers on OTB-100. Our method runs at the speed with 20FPS, which is faster than VITAL and CCOT. Overall, our tracker performs well against the state-of-the-art methods.

TABLE II
Speed Analysis

Trackers	VITAL	CCOT	ATOM	DaSiamRPN	MCPF	Ours
FPS	2	0 5	30	160	0 7	20

C. Qualitative Evaluation

Fig. 4 shows some tracking results of the high-performance trackers: GradNet, MCPF, LDES, ATOM and our method on 6 challenging sequences. In the proposed algorithm, the visual representations with CNN, CN and HOG features are effective. Our features contain both category-level semantics and spatial details, which account for appearance changes caused by deformation, rotation, illumination variation and background clutter (*MotorRolling*, *Skating2-1*, *Matrix*, *Ironman*, *Human9*, and *Board*). And for the most challenging *MotorRolling* sequence, none of the high-performance methods could track well whereas our method achieves the AUC score of 70.1%.

IV. Conclusion

This letter concurrently uses hand-crafted features with spatial details and deep features with semantics for visual tracking. In order to fuse these features using numerous observers and achieve high accuracy and resilience, the suggested tracker constructs a hybrid cascade structure. To further minimize computing cost, it can search for the target using a coarse-to-fine search strategy. Our algorithm's performance is demonstrated by experimental findings on two benchmark datasets, which compare it to the most advanced tracking algorithms available.

REFERENCES

[1] N. Wang, J. Shi, D. Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3101–3109.

[2] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 483–498.

[3] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2113–2120, 2015.

[4] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 263–270.

[5] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1401–1409.

[6] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3074–3082.

[7] X. Li, C. Ma, B. Wu, Z. He, and M. H. Yang, "Target-aware deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1369–1378.

[8] M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[9] T. Zhang, C. Xu, and M. H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4335–4343.

[10] C. Sun, D. Wang, H. Lu, and M. H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 489–497.

[11] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4670–4679.

[12] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1728–1740, Oct. 2008.

[13] N. Wang, S. Li, A. Gupta, and D. Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," 2015, *arXiv:1501.04587*.

[14] D. Bi, T. Ku, Y. Zha, L. Zhang, and Y. Yang, "Scale-adaptive object tracking based on color names histogram," *J. Electron. Inform. Technol.*, vol. 38, no. 5, pp. 1099–1106, May 2016.

[15] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2411–2418.

[16] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[17] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4660–4669.

[18] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6162–6171.

[19] Y. Li, J. Zhu, S. C. H. Hoi, W. Song, Z. Wang, and H. Liu, "Robust estimation of similarity transformation for visual object tracking," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 8666–8673, 2019.

[20] Y. Song *et al.*, "VITAL: Visual tracking via adversarial learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8990–8999.

[21] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–117.

[22] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 459–474.