Accelerated Particle Filter for Real-Time Visual Tracking With Decision Fusion

Shengjie Li, Shuai Zhao, Bo Cheng, *Member, IEEE*, and Junliang Chen

***Abstract*—Correlation-filter-based trackers, showing strong discrimination ability in challenging situations, have recently achieved superior performance in visual tracking. However, because the model treats the tracker's predictions in new frames as training data, the filter can be contaminated by small incorrect predictions, which cause model drift. Particle-filter-based trackers usually produce more accurate results due to the richer image representations used in prediction, but suffer when the environments are complex throughout an image sequence. In this letter, we propose an innovative real-time algorithm, which combines the particle filter with correlation filters in the prediction stage, enabling accurate predictions by the particle filter and alleviating model drift. Moreover, an effective decision fusion strategy is proposed to get more precise object predictions, thus further enhancing the overall tracking performance. Extensive evaluations on the OTB-2013 benchmark demonstrate that the proposed tracker is very promising compared with the state-of-the-art trackers, while operating over 85 frames/s.**

***Index Terms*—Correlation filter, decision fusion, particle filter, real time, visual tracking.**

I. Introduction

VISUAL tracking is one of the essential research topics in computer vision for its wide range of applications such as human–computer interaction, robotics, video surveillance, and driverless vehicles [1], [2]. The common setting for this problem is to initialize a target in the first frame with a bounding box. Then, the model-free online tracker needs to estimate the trajectory of this target throughout an image sequence. Although numerous trackers [3]–[9] have been proposed in the past decade, visual tracking is still a challenging problem due to cluttered backgrounds, fast motions, geometric deformations, illumination changes, and partial occlusions.

Manuscript received March 19, 2018; revised May 10, 2018; accepted May 30, 2018. Date of publication June 4, 2018; date of current version June 20, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61501048, in part by the Beijing Natural Science Foundation under Grant 4182042, in part by the Fundamental Research Funds for the Central Universities under Grant 2017RC12, and in part by the National Natural Science Foundation of China under Grant U1536111 and Grant U1536112. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mehdi Moradi. *(Corresponding author: Shengjie Li.)*

The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lishengjie@bupt.edu.cn; zhaoshuaiby@bupt.edu.cn; chengbo@bupt.edu.cn; chjl@bupt.edu.cn).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2018.2843524

Correlation-filter-based approaches have recently been applied to visual tracking. Benefited from the densely sampled strategy and quick operations in the Fourier domain, these trackers achieve high-speed performance and accurate predictions. An adaptive correlation filter, which learns the minimum output sum of the squared error, is first introduced by MOSSE [10] running at more than 600 frame/s (FPS) on the luminance channel. Some extensions, including a kernelized correlation filter [11], [12], scale estimation [13], [14], multiple channel features [12], [15], a regularized correlation filter [16], and a complementary learner [17], have been proposed to enhance the tracking performance. However, because these approaches treat the tracker's detection results in the following frames as training data, the detected and accumulated incorrect results would pollute the filter and cause model drift. Furthermore, the most widespread methods return the detection results through searching the maximum values of response maps produced by correlation filters.

The particle filter [18], using the richer image representations to produce more robust results, has been employed to tracking problems named Condensation [19]. The tracking performance is enhanced by increasing the number of particles. However, the tracking efficiency will be reduced due to the increasing computation. Some methods [20] have been proposed to speed up the particle filter. The subspace [21] is effectively represented by the Rao–Blackwellization particle filter, which draws fewer particles by analysis and calculation of the posterior over state vectors. An appearance-adaptive model [22] is integrated into the particle filter framework, where the number of particles depends on the noise variance. Unlike the previous approaches, the main difference in the particle filter framework lies in the observation model, in which correlation filters are adopted to reduce the number of particles.

The similar work can be found in [23], which proposes a multitask correlation filter to exploit fewer particles in the particle filter framework and performs favorably against the state-of-the-art methods. However, their real-time application is hindered by the high computational burden in the particle filter sampling process, where the features based on the convolutional neural networks (CNN) [24] are hierarchically applied with correlation filters [25], [26].

In this letter, we concern the issues mentioned above and propose an innovative accelerated particle filter visual tracking (APFT) approach, combining the particle filter with correlation filters. Unlike [23], we use the response map of the maximum weight particle, instead of considering the response maps of all particles, to locate the target with much fewer particles. Besides,

the APFT algorithm can make a necessary tradeoff between efficient and effective object tracking. The main contributions of this work are as follows.

1) We propose the accelerated particle filter, which integrates correlation filters with the particle filter using much fewer particles during the sampling process.
2) An effective decision fusion strategy is proposed to get more accurate object predictions, thus further boosting the overall tracking performance.
3) The evaluations on the OTB-2013 benchmark [28] show that the APFT tracker achieves a very promising performance compared with the state-of-the-art trackers, while operating in real time.

II. Proposed Algorithm

In this section, we first present the accelerated particle filter with the inherent scale estimation. Then, we introduce correlation filters integrated into the particle filter framework as a classifier. Finally, the decision fusion is presented to further enhance the ability to detect an object. Fig. 1 shows the APFT pipeline. The details are discussed in the following.

A. Accelerated Particle Filter

The tracking approach is formulated within a particle filter framework [29]. Let s^t and y^t represent the hidden state variables, which describe the affine parameters of the object and observation variable in the tth frame, respectively. Consequently, visual tracking can be modeled as the problem, in which all the observation variables up to the previous time $t-1$ are adopted to compute the most probable state variable

$$\begin{aligned} s^t &= \text{argmax}\, p(s^t|y^{1:t-1}) \\ &= \text{argmax} \int p(s^t|s^{t-1})p(s^{t-1}|y^{1:t-1})ds^{t-1}. \end{aligned} \tag{1}$$

When a new observation variable y^t arrives, we update the posterior distribution of s^t under Bayes' theorem

$$p(s^t|y^{1:t}) = \frac{p(y^t|s^t)p(s^t|y^{1:t-1})}{p(y^t|y^{1:t-1})}. \tag{2}$$

In the particle filter, the true posterior state distribution $p(s^t|y^{1:t})$ is approximated by a set of particles $\{s_i^t\}_{i=1}^n$ with corresponding normalized weights $\{\omega_i^t\}_{i=1}^n$. As failing to draw particles from the true posterior state distribution, an importance distribution $q(s^t|s^{1:t-1}, y^{1:t})$ is defined to approximate it. Hence, the weights are updated

$$\omega_i^t \propto \omega_i^{t-1} \cdot \frac{p(y^t|s_i^t)p(s_i^t|s_i^{t-1})}{q(s^t|s^{1:t-1}, y^{1:t})}. \tag{3}$$

In particle-filter-based trackers, a first-order Markov process $q(s^t|s^{t-1})$ is often used to describe the importance distribution $q(s^t|s^{1:t-1}, y^{1:t})$, thus leading to $\omega_i^t = \omega_i^{t-1} p(y^t|s_i^t)$. As the degeneracy problem [29] exists, the resampling strategy is then employed during the update phase, and their weights are reset to $1/n$. Generally, the state parameters s_i of the target are denoted by six affine transformation variables: translation in the x-axis, translation in the y-axis, scale, rotation, aspect ratio, and skew angle. These parameters of $q(s^t|s^{t-1})$ are independent and assumed by a Gaussian distribution.

The main difference in existing particle-filter-based tracking algorithms [30], [31] is in the observation models $p(y^t|s_i^t)$ defined as the probability corresponding to the classifier output. Different from these methods, the accelerated particle filter formulates the maximum values of the response maps, which are generated by correlation filters equivalent to a classifier, as the particle weights similar to [23]

$$\omega_i = \max(g^i). \tag{4}$$

Here, g represents response maps and i denotes the ith particle. Unlike [23], the object position can be predicted by searching the maximum value of $\boldsymbol{\omega}$ with much fewer particles

$$(s^t|y^{1:t})_{\text{state}} = s^t_{\max(\boldsymbol{\omega})}. \tag{5}$$

B. Correlation Filter

Multiple correlation filters are integrated into our tracking approach for the decision fusion. We train the kth correlation filter [11]–[13], [15], [32] on the features of target patch $\boldsymbol{\varphi}(\boldsymbol{x}) = [\varphi^1(\boldsymbol{x}), \varphi^2(\boldsymbol{x}), ..., \varphi^l(\boldsymbol{x})] \in \mathbb{R}^{M\times N\times D}$. l means multiple channels of features. Many effective features are allowed to integrate. The ideal scores $\boldsymbol{y} \in \mathbb{R}^{M\times N}$ are the Gaussian function decaying from 1 to 0. We obtain the desired filter $\boldsymbol{\alpha}$ by minimizing the loss function of ridge regression

$$\varepsilon = \| \boldsymbol{\alpha} \star \boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{y} \|^2 + \lambda \| \boldsymbol{\alpha} \|^2 \tag{6}$$

where $\star$ represents circular correlation and $\lambda > 0$ is a regularization coefficient. The closed-form solution is obtained as follows [13]:

$$\hat{\boldsymbol{\alpha}}^l = \frac{\hat{\varphi^l(\boldsymbol{x})} \odot \hat{\boldsymbol{y}}^*}{\sum_{l=1}^D \hat{\varphi^l(\boldsymbol{x})} \odot \hat{\varphi^l(\boldsymbol{x})}^* + \lambda}. \tag{7}$$

Here, the hats in $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{\alpha}}$, and $\hat{\varphi(\boldsymbol{x})}$ denote the discrete Fourier operator $\mathcal{F}$, $\odot$ represents the Hadamard product, and $\hat{\boldsymbol{y}}^*$ is the complex conjugate of $\hat{\boldsymbol{y}}$.

During the prediction stage, we draw the samples $\boldsymbol{\varphi}(\boldsymbol{z^i})$ by the particle filter at the previous location, and the peak of the kth response map about the ith sample is computed in the Fourier domain

$$g_k^i = \mathcal{F}^{-1}\left(\sum_{l=1}^D \hat{\boldsymbol{\alpha}}^{\boldsymbol{l}*} \odot \hat{\boldsymbol{\varphi}^{\boldsymbol{l}}(\boldsymbol{z^i})}\right). \tag{8}$$

C. Decision Fusion

Various features can be employed to exploit the advantages of the decision fusion. We adopt three commonly used features [histogram of oriented gradient (HOG) and color names (CN)/intensity channels (IC)] in the proposed approach. Different from the existing multiple feature integration [14], [33], we integrate features at the decision-making level of response maps for more accurate detection and alleviating model drift. Given the set of response maps g_k^i, we fuse these maps produced

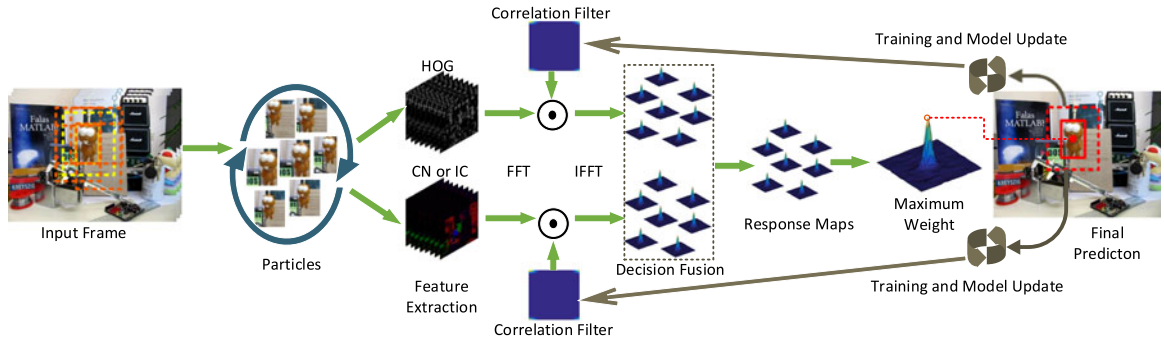

Fig. 1. Flowchart of the proposed APFT for real-time visual tracking. The yellow and red dashed lines represent the padding box. The orange dotted line is the box of drawn samples. The solid yellow line is the input bounding box, and the solid red line represents the predicted bounding box. CN [15] is used for the color sequences and IC [27] is used for the gray sequences.

by correlation filters with different features. Let g^i indicate the fused response maps on the ith particle, which is formulated as

$$g^i = \sum_{k=1}^{K} (\mu_k * g_k^i)$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \mu_k = 1. \tag{9}$$

Note that translation variances in each state s_i^t are forced to move to the predicted position according to the peak of the fused response map for the match between states and weights, before the final object location is predicted. Although this operation is quite straightforward, the performance of APFT using much fewer particles is very promising.

D. Tracking and Model Update

The APFT tracker is proposed based on the particle filter and correlation filters, as shown in Fig. 1. First, the model $p(s^t|s^{t-1})$ is applied to draw samples and resample. Next, correlation filters are adopted to compute response maps on different features of each sample. Then, the decision fusion is employed to integrate these response maps, and the weights are updated by using the peaks of the fused response maps. Finally, the estimated state corresponding to the maximum weight is the target position.

Note that we fix all samples to the same size for computing response maps; then, the model update is similar to the correlation filters [11], [12]. The kth filter $\tilde{\boldsymbol{\alpha}}$ and the kth learned target feature $\boldsymbol{\varphi}(\tilde{\boldsymbol{x}})$ are updated, respectively, by the learning rate parameter θ:

$$\mathcal{F}(\tilde{\boldsymbol{\alpha}}) = (1-\theta)\mathcal{F}(\tilde{\boldsymbol{\alpha}}) + \theta\mathcal{F}(\boldsymbol{\alpha})^{\text{new}} \tag{10}$$
$$\mathcal{F}(\boldsymbol{\varphi}(\tilde{\boldsymbol{x}})) = (1-\theta)\mathcal{F}(\boldsymbol{\varphi}(\tilde{\boldsymbol{x}})) + \theta\mathcal{F}(\boldsymbol{\varphi}(\boldsymbol{x}))^{\text{new}}. \tag{11}$$

III. Experiments

To fully validate the effectiveness of the APFT tracker, we execute experiments on the OTB-2013 dataset [28], which consists of 51 video sequences. According to [28], we apply two evaluation criterions: precision scores and success scores. The former shows the rate of frames, whose center location is within 20 pixels of ground-truth positions, while the latter is reported as the area under the curve of the success plot, which is the overlap percentage between the tracking results and ground truth boxes. Furthermore, in order to verify the robustness of APFT, we take three metrics proposed in [28], which are one-pass evaluation (OPE), spatial robustness evaluation (SRE), and temporal robustness evaluation (TRE). More results can be seen in the online supplementary material.

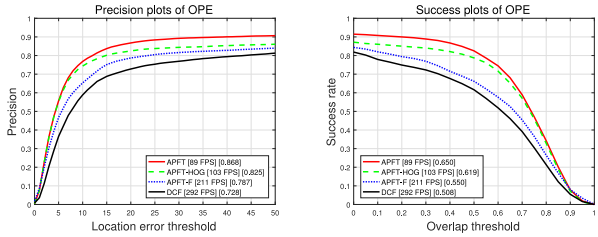

Fig. 2. Comparisons of the component effectiveness on OTB-2013 in OPE. DCF is reported in [12]. APFT-HOG is developed with HOG features. APFT-F is implemented without the affine parameters, which are set to [0, 0, 0, 0, 0, 0]. We can see that APFT achieves the best performance among them. The FPS is presented for the tracking efficiency analysis.

A. Implementation Details

The proposed tracker is executed in MATLAB without any optimization. All the experiments are implemented on a PC with an Intel Core i7-7700HQ CPU (2.80 GHz), 8-GB RAM. APFT runs at about 89 FPS using HOG and CN/IC, of which the fusion parameter μ_k is set to [0.75 0.25]. The setting of HOG features is similar to [12]. The regularization parameter λ is set to 0.0001. The learning rate θ is set to 0.01. As in [34], the affine parameters for the state variable are set to [6, 6, 0.031, 0, 0.001, 0]. The number of particles is set to 5.

B. Component Analysis of APFT

In this section, we instruct the component analysis with the precision and success scores on OTB-2013 in OPE. Theoretically, our tracking performance is primarily related to the decision fusion and the particle filter, thus implementing three extra algorithms to demonstrate the effective tracking performance of APFT. First, we apply DCF [12] as the baseline algorithm, which is similar to APFT without the particle filter and the

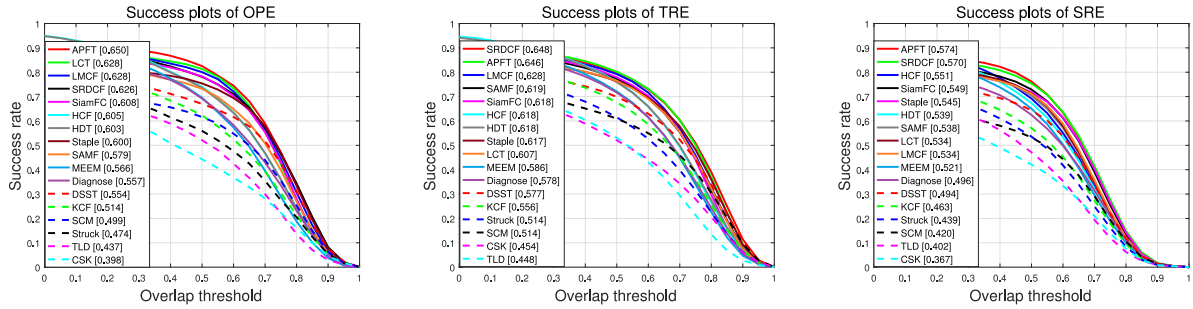

Fig. 3. Success plots on OTB-2013 corresponding to OPE, TRE and SRE. the proposed APFT algorithm has a very promising performance compared with the state-of-the-art trackers.

decision fusion. We then construct APFT-F by using the decision fusion without the particle filter. Finally, APFT-HOG is implemented by using HOG features and the particle filter. Fig. 2 shows the results of the precision and success scores in OPE. According to Fig. 2, APFT-F with the decision fusion achieves about 5.9% and 4.2% improvement in the precision and success scores compared to DCF, respectively. Although the decision fusion is not applied in APFT-HOG, the precision and success scores of APFT-HOG adopting the particle filter are 3.8% and 6.9% better than that of APFT-F, respectively. As is shown, APFT performs favorably against APFT-HOG with about 4.3% and 3.1% improvement in the precision and success scores attributed to the decision fusion, respectively. The notable result is that compared with the baseline algorithm, APFT achieves 14% and 14.2% improvement in the precision and success scores, respectively.

C. *Efficiency and Particle Number Analysis of APFT*

In this section, we discuss the effect of the number of particles on the success scores about OTB-2013 in OPE. By moving the translation variances according to the peaks of the fused response maps, APFT can use much fewer particles to locate the target. As shown in Fig. 3, even using one particle only, the proposed APFT method can achieve the competitive performance with the highest 211 FPS. Besides, the performance improvement is obtained by increasing the number of particles, but the runtime performance suffers. We set the reasonable number of particles to five, which can effectively make a tradeoff between accuracy and efficiency of our tracker.

D. *Comparison With State-of-the-Art Trackers*

To further validate the effectiveness and robustness of APFT, we implement APFT with comparisons to 16 state-of-the-art trackers:

1) correlation trackers (SRDCF [16], LCT [32], Staple [17], HCF [25], HDT [35], SAMF [14], CSK [11], KCF [12], DSST [13], and LMCF [33]);
2) particle-filter trackers (Diagnose [36] and SCM [5]);
3) other trackers (SiamFC [37], MEEM [38], Struck [39], and TLD [6]).

From Fig. 4, we see that APFT is very competitive to correlation-filter-based trackers in all evaluations. Although the success score of APFT in TRE is 0.2% lower than that of

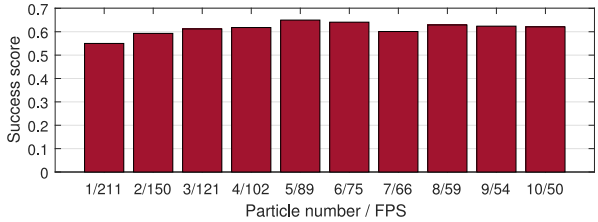

Fig. 4. Evaluating APFT with respect to the different number of particles. The success score in OPE over the OTB-2013 dataset is displayed. One particle provides inferior performance operating at the highest 211 FPS. The performance then improves with increasing particles, until five particles. The performance that is sensitive to affine parameters then fluctuates for more particles, while the FPS decreases. We set five particles for all experiments.

SRDCF, the success scores of AFPT are 2.4% and 0.4% better than that of SRDCF in OPE and SRE, respectively. In terms of tracking speeds, LMCF and KCF are reported about 80 and 172 FPS, respectively, while APFT outperforms them in all success scores.

Compared with particle-filter-based methods including SCM and Diagnose, the APFT tracker has about 15.1% and 9.3% improvement with the success scores in OPE, respectively. Furthermore, the proposed APFT runs over 160 times faster than SCM (about 0.4 FPS) and over eight times faster than Diagnose (about 10 FPS).

Compared to HCF and HDT using CNN features without handling scale change, even if APFT just adopts hand-crafted features, the performance gains of APFT are 4.5% and 4.7% in terms of the success scores in OPE, respectively. Overall, the proposed APFT outperforms all other trackers with the success scores in OPE and SRE.

IV. Conclusion

In this letter, we propose an innovative accelerated particle filter for real-time visual tracking. Correlation filters in the APFT tracker can reduce the number of particles in the particle filter, while the particle filter in the APFT tracker can improve the robustness of correlation filters. Moreover, in contrast to the conventional correlation-filter-based trackers, the APFT tacker adopting the particle filter and the decision fusion produces more accurate predictions to efficiently alleviate model drift. Overall, the APFT tracker is very promising in the performance compared with the state-of-the-art trackers, while running in real time.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006, Art. no. 13.

[2] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Deghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1142–1468, Jul. 2014.

[3] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1822–1829.

[4] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[5] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1838–1845.

[6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[7] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1269–1276.

[8] S. Zhang, S. Kasiviswanathan, P. Yuen, and M. Harandi, "Online dictionary learning on symmetric positive definite manifolds with vision applications," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3165–3173.

[9] S. Zhang, H. Zhou, H. Yao, Y. Zhang, K. Wang, and J. Zhang, "Adaptive NormalHedge for robust visual tracking," *Signal Process.*, vol. 110, pp. 132–142, 2015.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2544–2550.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[13] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–5.

[14] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2014, pp. 254–265.

[15] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1090–1097.

[16] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4310–4318.

[17] 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, vol. 38, pp. 1401–1409.

[18] A. Doucet, D. N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001.

[19] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, pp. 5–28, 1998.

[20] C.Yang, R. Duraiswami and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 212–219.

[21] Z. Khan, T. Balch, and F. Dellaert, "A Rao-Blackwellized particle filter for eigentracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2004, pp. 980–986.

[22] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1419–1506, Nov. 2004.

[23] 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. 4819–4827.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[25] 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.

[26] Y. Qi *et al*., "Hedging deep features for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018, to be published.

[27] M. Felsberg, "Enhanced distribution field tracking using channel representations," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 121–128.

[28] 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.

[29] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[30] S. Zhang, H. Zhou, F. Jiang, and X. Li, "Robust visual tracking using structurally random projection and weighted least squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1749–1760, Nov. 2015.

[31] S. Zhang, X. Lan, H. Yao, H. Zhou, D. Tao, and X. Li, "A biologically inspired appearance model for robust visual tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2357–2370, Oct. 2016.

[32] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5388–5396.

[33] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4800–4808.

[34] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 125–141, 2008.

[35] Y. Qi *et al.* "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4303–4311.

[36] 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.

[37] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 850–865.

[38] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, vol. 8694, pp. 188–203.

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