Motion Correlation Discovery for Visual Tracking

Weijun Zhang, Sheng Zhong, Wenhui Xu, and Ying Wu, *Fellow, IEEE*

***Abstract*—Motion information plays an important role in identifying moving objects, which has not been well utilized in state-of-the-art tracking algorithms. In this letter, we propose a unified framework integrating two tracking problems, i.e., pixel-level foreground probabilistic inference and motion parameter estimation. Our model employs motion fields to propagate probability forward, and discovers motion patterns in the spatial domain to distinguish targets from the background. It takes advantage of continuity and inertia of both target and camera motion, and provides reliable evidence to resolve confusion caused by appearance similarity between targets and the background. Target localization is effectively achieved from the pixel-level foreground probabilistic map. Experimental results demonstrate that the proposed method significantly improves our baseline method, and achieves performance comparable to state-of-the-art tracking methods with more complex features.**

***Index Terms*—Visual tracking, motion analysis, pixel-level probabilistic model, Bayesian inference.**

I. Introduction

VISUAL tracking is a fundamental research problem in video analysis and is demanded by many contemporary applications. It is a challenging task to perform model-free tracking as no prior about the target is available, and appearance is usually the only observation given, which could be non-stationary in some situations. While recent years have witnessed a steady advance in both theory and practice [1]–[3], target representation remains a challenging problem.

Rectangle template bounding the target region is quite a popular target representation, adopted by both generative models [4]–[6] and discriminative models [2], [7], [8], with state-of-the-art performance achieved. However, by including some background clutter as part of the target representation, the model is quite likely to drift away from the target gradually because of noise and error accumulation, especially in challenge situations where target deforms or moves fast.

The human vision system does not identify the target with a rectangle template as we know exactly which pixels belong to the target, and have the knowledge that pixels belonging to the same object share the same motion trend. Research from cognition and psychology community [9], [10] shows that several-month-old human infants already have the knowledge that a hidden, freely moving object moves continuously and smoothly. This human knowledge about motion, however, has not been modeled into existing visual tracking methods.

To integrate these motion cues, we formulate a pixel-level foreground probabilistic inference problem. A motion field provides pixel-level motion correlation between successive frames, and is used to propagate a probabilistic map forward. In addition, by taking advantage of continuity and inertia of both target and camera motion, we discover discriminative motion patterns to distinguish targets from the background. These together with image observation from other features are combined into a unified Bayesian framework, and pixel-level foreground probabilistic inference is produced recursively.

II. Related Work

Some recent works have attempted to build pixel-level models. Segmentation-based trackers aim to distinguish target and background pixels by using various kinds of segmentation techniques (e.g., image matting [11], grab cut [12], level sets [13], [14]). Foreground probabilistic inference methods go one-step further. ELK tracking [15] builds probabilistic inference into an extended Lucas-Kande framework. Other methods employ various techniques like color histograms [16], gradient boosting decision trees [17] and generalized Hough voting [18] to generate probabilistic segmentation.

In [19]–[21] tracking and segmentation are simultaneously addressed by applying Bayesian inference to fuse pixel-wise cues from different sources. While these methods are closely related to our Bayesian framework, their transition probabilities are modeled in different manners, as well as pixel-wise likelihoods. Bayesian inference is also widely used in multiple-object estimation [22], recognition [23] and tracking [24] problems.

A recent trend in visual tracking is the use of more complex features, especially pre-trained [25], [26] and task-specific [27], [28] deep features. [29] introduces motion cues into tracking-by-detection framework by incorporating deep motion features, which holds the same view as us that motion cues provide discriminative and complementary information to existing methods.

Staple [30] observes that pixel-level models [12], [16], [18] are good at handling deformable targets, while template-based models [1], [7], [8] are robust for tracking rigid objects. As these two models provide information complementary to each other, a simple combination of them achieves favorable performance against each part. In our work, we adopt the same tracking

Manuscript received June 28, 2018; revised September 4, 2018; accepted September 24, 2018. Date of publication September 28, 2018; date of current version October 5, 2018. This work was supported by the National Key R&D Program of China under Grant 2016YFF0101502. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vasileios Mezaris. *(Corresponding author: Weijun Zhang.)*

W. Zhang, S. Zhong, and W. Xu are with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: weijunzhang@hust.edu.cn; shengzhong@hust.edu.cn; xuwenhui@hust.edu.cn).

Y. Wu is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208-3118 USA (e-mail: yingwu@ece.northwestern.edu).

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.2018.2872679

procedure as [30], and demonstrate how this baseline tracker is improved by incorporating motion cues into the pixel-level model.

III. Proposed Tracking Method

A. Overall Tracking Procedure

We adopt the tracking-by-detection paradigm, in which, in frame t, the rectangle p_t that gives the target location in image y_t is chosen from a set S_t to maximize a score:

$$p_t = \arg\max_{p \in S_t} f(T(y_t, p); \theta_{t-1}). \tag{1}$$

The function T is an image transformation such that $f(T(y, p); \theta)$ assigns a score to the rectangular window p in image y according to the model parameters θ. The model parameters θ_{t-1} should be trained with the previous images and the location of the object in those images $\{(y_i, p_i)\}_{i=1}^{t-1}$.

We adopt a score function that is a linear combination of scores from template-based and pixel-level models:

$$f(x) = \gamma f_{tmpl}(x) + (1 - \gamma) f_{pxl}(x). \tag{2}$$

Coefficient γ is used to balance the two components.

The template-based response $f_{tmpl}(x)$ is calculated under a Correlation Filter formulation [7]. The pixel-level response is computed from an M-channel feature image $\psi_x : \mathcal{H} \rightarrow \mathbb{R}^M$, obtained from x and is defined on a finite grid $\mathcal{H} \subset \mathbb{Z}^2$:

$$f_{pxl}(x; \beta) = g(\psi_x; \beta) = \frac{1}{|\mathcal{H}|} \sum_{u \in \mathcal{H}} \zeta(\beta, \psi_x)[u], \tag{3}$$

which is the average of a scalar-valued score image $\zeta_{(\beta, \psi_x)}$ on the finite grid $\mathcal{H}$. The score image $\zeta_{(\beta, \psi_x)}$ is calculated on $\Omega \subset \mathbb{Z}^2$, a search area centered at the position in the previous image p_{t-1} (red rectangle in Fig. 1). Then we obtain $f_{pxl}(x; \beta)$ by calculating an integral image on $\zeta_{(\beta, \psi_x)}$.

The above mentioned framework is the same as Staple [30]. In [30] $\zeta_{(\beta, \psi_x)}$ is calculated from simple color histograms. In our method it is a pixel-level foreground probabilistic map inferred from a Bayesian inference framework recursively collecting color-based and motion-based cues. Fig. 1 shows the overall tracking procedure of our method, and orange arrows highlight additional components compared to [30].

B. Pixel-Level Bayesian Inference Formulation

Let $c_x^t \in \{0, 1\}$ and y_x^t be the class (with $c_x^t = 1$ indicating target and otherwise background) and image observation of pixel at position $x \in \Omega$ at time t, and $y_x^{1:t}$ be set of image observations from y_x^1 to y_x^t. For every pixel location $x = [u_x, v_x]^\mathsf{T}$ in our region of interest at time t, the probability of x belonging to class $C \in \{0, 1\}$ is inferred according to image observations up to the current frame. We have a recursive Bayesian formulation

$$p(c_x^t = C | y^{1:t-1}) = \int p(c_x^t = C | c_{x'}^{t-1}) p(c_{x'}^{t-1} | y^{1:t-1}) dc_{x'}^{t-1} \tag{4}$$

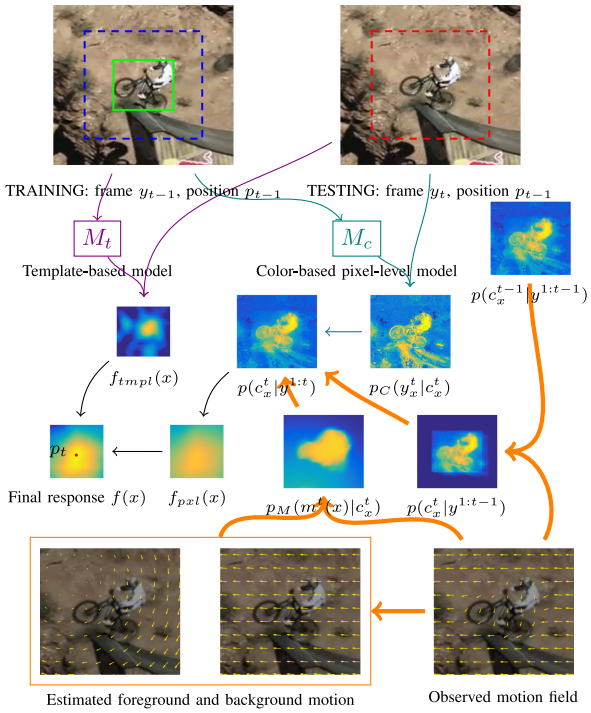

Fig. 1. Overall tracking procedure. Template-related. In frame y_{t-1}, a training patch (blue rectangle) represented using HOG features is extracted at the estimated location p_{t-1} and used to update parameters of the template-based model. In frame y_t, we exact features for the testing patch around the location in the previous image p_{t-1} (red rectangle) and use the previously trained model to obtain the dense template response. Color-related. In frame y_{t-1}, foreground (green rectangle) and surrounded background regions (blue rectangle) are used to update the model in (7), (8). In frame y_t, a per-pixel score is computed in a search area centered at the position in the previous image (red rectangle) using the color-based model. Motion-related. With our dynamic model, a new foreground probabilistic map is predicted according to the probabilistic map from last frame and observed motion field using (4). With our motion-based observation model, foreground and background motion patterns are estimated, and further used to calculate a motion-based likelihood map using (9) to (15). All the color-based and motion-based maps are combined to form a probabilistic map $p(c_x^t|y^{1:t})$ by our Bayesian inference framework (5), (6), which is then used to compute the dense response efficiently using an integral image (3). The final response $f(x)$ is obtained with (2) and the new location p_t of the target is estimated at its peak with (1). Best viewed in color.

where x' is the corresponding position at $t-1$ of position x at time t, and

$$p(c_x^t = C | y^{1:t}) \propto p(y_x^t | c_x^t = C) p(c_x^t = C | y^{1:t-1}). \tag{5}$$

In (4) a new foreground probabilistic map is predicted according to the probabilistic map from last frame, and transition probability between frames $p(c_x^t | c_{x'}^{t-1})$. In (5) per-pixel image observation from current frame $p(y_x^t | c_x^t)$ is combined to correct the estimation. It consists of two parts that are assumed to be independent of each other, a color-based model and a motion-based model, i.e.,

$$p(y_x^t | c_x^t) \approx p_C(y_x^t | c_x^t) p_M(m^t(x) | c_x^t) \tag{6}$$

where $m^t(x)$ is backward optical flow observation at pixel location x at time t.

Our formulation models motion cues in two ways. Temporal motion correlation is modeled by $p(c_x^t | c_{x'}^{t-1})$, described in Section III-C, and spatial motion correlation is captured by

$p_M(m^t(x)|c_x^t)$, described in Section III-E. In addition, we adopt the color-based likelihood $p_C(y_x^t|c_x^t)$ in our baseline method [30] as part of our observation model, since it is reliable when the target area has a discriminative appearance characteristic. We give details about this term in Section III-D.

C. Dynamic Model

Pixel observation matching gives correlation between pixels from successive frames, and is achieved by a state-of-the-art optical flow method [31], which gives a matching field at sub-pixel level. Given backward optical flow $m^t(x)$ at pixel location x at time t, a pixel observation at location x is matched to $x' = x + m^t(x)$ in time $t-1$. As x' could be a sub-pixel location, and for region $x \in \Omega$ we already have the foreground probabilistic map $p(c_x^{t-1}|y^{1:t-1})$, foreground probability $p(c_{x'}^{t-1}|y^{1:t-1})$ is obtained via interpolation. The transition probability $p(c_x^t = C|c_{x'}^{t-1} = C)$ represents fidelity to previous estimated probabilistic map and should be higher than 0.5. We find that values in $[0.8, 1]$ make little difference, and set it as 1 (highest fidelity) in our experiments.

Note that recent work [21] also uses a pixel-level recursive Bayesian formulation. Their formulation, however, avoids the modeling of pixel matching between successive frames, and assumes pixel locations to be independent to simplify computational complexity, which actually works well in the cases where the frame rate is relatively high and hence the target motion very slow, but limits its generalizability to more complex scenarios where the target deforms or moves fast.

D. Color-Based Observation Model

A histogram based Bayes classifier is employed, with RGB color histograms using 32 bins per channel. Let $H_1^t(b)$ and $H_0^t(b)$ denote the b-th bin of the normalized foreground and background histograms computed over search region Ω at time t, and let Ω_b be set of pixels with features corresponding to the b-th bin. In testing step, we estimate $p_C(y_x^t|c_x^t = C)$ through $H_C^{t-1}(b)$. In learning step, the histograms are computed accounting pixel level foreground probability as

$$\tilde{H}_C^t(b) = \sum_{x \in \Omega_b} p(c_x^t = C) \Big/ \sum_{x \in \Omega} P(c_x^t = C). \tag{7}$$

The model is updated online with a learning rate η_{hist} as

$$H_C^t(b) = (1 - \eta_{hist}) H_C^{t-1}(b) + \eta_{hist} \tilde{H}_C^t(b). \tag{8}$$

E. Motion-Based Observation Model

This idea is based on the observation that pixels belong to the same object share the same motion trend, especially in fast moving circumstances, which is helpful for resolving signals raised by background clutter. In Fig. 1 the background has similar appearance to the target, but clearly presents a different motion trend, which is an informative observation for distinguishing background clutter from the target.

We aim to estimate the rotation angles θ_C and translation vectors $[u_C, v_C]^\mathsf{T}$ in every frame, with $C \in \{0, 1\}$ (with 0 representing background and 1 representing target), then we have motion state parameters $s = [\theta_0, u_0, v_0, \theta_1, u_1, v_1]$.

Given s, Let A_1 and b_1 be rotation matrix and translation vector of foreground, A_0 and b_0 be those of surrounding background, i.e.,

$$A_C(s) = \begin{bmatrix} \cos(\theta_C) & -\sin(\theta_C) \\ \sin(\theta_C) & \cos(\theta_C) \end{bmatrix}, b_C(s) = \begin{bmatrix} u_C \\ v_C \end{bmatrix}. \tag{9}$$

Under the assumption of x being foreground/background location, with known motion s, the ideal motion vector of pixel at location x, is given by $M_C(x, s) = A_C(s)x + b_C(s) - x$. The observed motion vector $m(x)$ is a noisy observation of the ideal motion vector, and we model motion components at both directions as two scalar Gaussian distributions, i.e.,

$$p(m(x)|c_x = C, s = S) = \mathcal{N}(m(x)|M_C(x, S), \Sigma_V) \tag{10}$$

where $\Sigma_V = diag(\sigma_{u_C}, \sigma_{v_C})$ is a diagonal covariance matrix whose elements are the variances of components of the motion vector. Examples of $M_1(x, s)$, $M_0(x, s)$ and $m(x)$ are visualized in the last row of Fig. 1.

We then estimate the motion-based pixel level likelihood as

$$p_M(m(x)|c_x = C) \approx p(m(x)|c_x = C, s = \hat{s}) \tag{11}$$

where $\hat{s}$ is the best estimation of s. Note that in (9) to (11) we omit superscript t for conciseness.

Motion state s^t at time t is sequentially estimated by a particle filter framework, given $z^t = m^t(x)$ as motion observation at time t. Given the observation set $\mathcal{Z}^t = \{z^i\}_{i=1}^t$ up to frame t, the motion state s^t is obtained by maximizing a posteriori probability

$$\hat{s}^t = \arg\max_{s^t} \{p(s^t|\mathcal{Z}^t) \propto p(z^t|s^t) p(s^t|\mathcal{Z}^{t-1})\} \tag{12}$$

where $p(s^t|\mathcal{Z}^{t-1}) = \int p(s^t|s^{t-1}) P(s^{t-1}|\mathcal{Z}^{t-1}) ds^{t-1}$, and $p(s^t|s^{t-1})$ is a dynamic model that describes the temporal correlation of the target states in two consecutive frames and $p(z^t|s^t)$ is the observation model that denotes the likelihood of an observation given a state. We assume that the state parameters are independent and are modeled by six scalar Gaussian distributions between two consecutive frames, i.e., $p(s^t|s^{t-1}) = \mathcal{N}(s^t|s^{t-1}, \Sigma_M)$, where $\Sigma_M = diag(\sigma'_{\theta_0}, \sigma'_{u_0}, \sigma'_{v_0}, \sigma'_{\theta_1}, \sigma'_{u_1}, \sigma'_{v_1})$ is a diagonal covariance matrix whose elements are the variances of the motion parameters. In visual tracking, the posterior probability $p(s^t|Z^t)$ in (12) is approximated by a set of particles $\{s_i^t\}_{i=1}^{n_p}$ that are sampled with their corresponding importance weights $\{w_i^t\}_{i=1}^{n_p}$, where $w_i^t \propto p(z^t|s_i^t)$. Therefore, (12) can be approximated as

$$\hat{s}^t = \arg \max_{\{s_i^t\}_{i=1}^{n_p}} p(z^t|s_i^t) p(s_i^t|\hat{s}^{t-1}). \tag{13}$$

The observation model $p(z^t|s^t)$ denotes the likelihood of an observation given a state, and is estimated from likelihood of the motion field as

$$p(z^t|s_i^t) \propto \int_\Omega p(m^t(x)|s = s_i^t) dx \tag{14}$$

where the probability of observing $m^t(x)$ given s_i^t is

$$p(m^t(x)|s = s_i^t) = \int p(m^t(x)|c_x^t = C, s = s_i^t) p(c_x^t = C) dc_x^t. \tag{15}$$

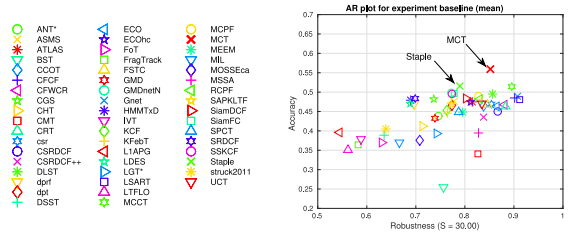

Fig. 2. Accuracy-Robustness plots on VOT2017 benchmark. Better trackers are closer to the top right corner.

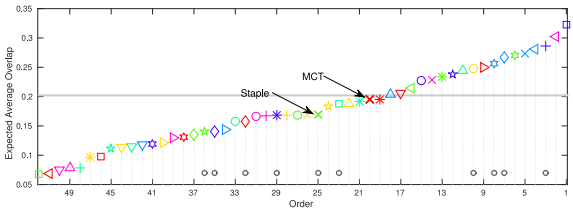

Fig. 3. Expected average overlap graph with trackers ranked from right to left. The dashed horizontal line denotes the average performance of ten state-of-the-art trackers published in 2016 and 2017 at major computer vision venues (denoted by gray circles in the bottom part of the graph), suggested by VOT as state-of-the-art bound.

Here the tracking problem of (13) to (15) aims to estimate the global motion state of the target and background $\hat{s}$, and is complementary and related to the pixel level probability estimation problem (4) and (5), which gives a probability map $p(c_x^t = C|y^{1:t})$.

The motion state parameters $\hat{s}$ are required for solving the pixel level probability estimation problem, so $p(c_x^t = C|y^{1:t})$ is not available yet when estimating $\hat{s}$ through (13) to (15). To resolve this chicken and egg dilemma, we use $p(c_x^t = C) \approx p(c_x^t = C|y^{1:t-1})$ in (15) as approximation.

IV. Experiments

We compare our MCT (Motion Correlation Tracker) to competing methods on two recent and popular benchmarks, VOT2017 [3] and OTB100 [32], and demonstrate performance comparable to state-of-the-art. All parameters are set to values same as our baseline Staple [30]. All parameters have a straightforward interpretation, do not require fine-tuning, and are kept constant throughout all experiments. Our Matlab implementation runs at 23 frames per second on an Intel Core i5 2.5 GHz standard desktop.

A. Experiments on VOT2017

The VOT2017 [3] benchmark contains results of 51 state-of-the-art trackers evaluated on 60 challenging sequences. The VOT methodology resets a tracker upon failure to fully use the dataset. Three primary measures are the number of failures during tracking (robustness), average overlap during the periods of successful tracking (accuracy), and expected average overlap (EAO) [33] estimated for a selected range of sequence lengths.

Fig. 2 visualizes two independent measures, i.e., accuracy and robustness on two axes. Our method significantly improves our baseline method Staple [30] in terms of both measures, and is by far the best method in terms of accuracy. Fig. 3 shows EAO graph with trackers ranked from right to left. Our tracker promotes EAO score of our baseline from 17% to 20%, and achieves performance comparable to state-of-the-art (dashed horizontal line on the graph) according to the VOT standards.

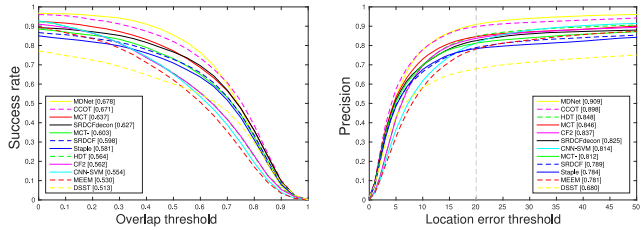

Fig. 4. One-Pass-Evaluation (OPE) curves on OTB100 dataset. **Left**: Success plot. Trackers are ranked using area under the curve (AUC). **Right**: Precision plot. Precision score with threshold equal to 20 pixels is used to rank the trackers.

B. Experiments on OTB100

The OTB100 [32] benchmark contains results of 29 trackers evaluated on 100 sequences by a no-reset evaluation protocol. Performance is measured by precision (center location error between ground truth and reported bounding box) and success rate (the percentage of frames where overlap with the ground truth is higher than a threshold).

We compare our method to representative state-of-the-art trackers. The success plot in Fig. 4 shows that our method not only outperforms simple and fast trackers like DSST [7], Staple [30], but also outperforms trackers using deep features like CF2 [27], CNN-SVM [34], HDT [35], and trackers that use more complex models and run far slower like MEEM [36], SRDCF [2] and SRDCFdecon [37]. We also include top-performance trackers like MDNet [38] and CCOT [25] which run much slower than MCT in our experiments. The precision plot demonstrates a similar result.

As motion cues are combined by adding two components, i.e., temporal and spatial motion correlation into our baseline Staple [30], we also include a tracker adding only temporal motion correlation in our experiments, marked as MCT- in Fig. 4. Both measures show that MCT- improves Staple, and MCT improves MCT-, meaning that both components introduce discriminative and complementary information.

Videos in OTB100 are annotated with 11 attributes indicating challenging situations. We find that MCT fails to improve the baseline in the cases of occlusion and motion blur because of difficulty in estimating the correct motion fields.

V. Conclusion

In this letter we formulate visual tracking as a pixel-level foreground probabilistic inference problem, into which motion cues are combined in two ways. Temporal motion correlation between pixels in successive frames is modeled as transition probability in our Bayesian network. Spatial motion correlation between pixels belonging to the same object is learned by estimating both target and camera motion parameters, and serves as the observation model. Our formulation provides an elegant tracking framework integrating motion, color and structure cues, achieves performance comparable to state-of-the-art algorithms, and produces a foreground probabilistic map and motion parameters as by-products.

REFERENCES

[1] 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.

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

[3] M. Kristan *et al.*, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct. 2017, pp. 1949–1972.

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

[5] 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.

[6] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 470–484.

[7] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Nottingham, U.K., Sep. 1–5, 2014, pp. 1–11.

[8] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[9] E. S. Spelke, G. Katz, S. E. Purcell, S. M. Ehrlich, and K. Breinlinger, "Early knowledge of object motion: Continuity and inertia," *Cognition*, vol. 51, no. 2, pp. 131–176, 1994.

[10] S. P. Johnson and R. N. Aslin, "Perception of object unity in young infants: The roles of motion, depth, and orientation," *Cogn. Develop.*, vol. 11, no. 2, pp. 161–180, 1996.

[11] J. Fan, X. Shen, and Y. Wu, "Scribble tracker: A matting-based approach for robust tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1633–1644, Aug. 2012.

[12] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Comput. Vis. Image Understanding*, vol. 117, no. 10, pp. 1245–1256, 2013.

[13] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 831–844.

[14] V. A. Prisacariu and I. Reid, "Nonlinear shape manifolds as shape priors in level set segmentation and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2185–2192.

[15] S. Oron, A. Bar-Hille, and S. Avidan, "Extended Lucas-Kanade tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 142–156.

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

[17] J. Son, I. Jung, K. Park, and B. Han, "Tracking-by-segmentation with online gradient boosting decision tree," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3056–3064.

[18] S. Duffner and C. Garcia, "Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2480–2487.

[19] C. Aeschliman, J. Park, and A. C. Kak, "A probabilistic framework for joint segmentation and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1371–1378.

[20] K. E. Papoutsakis and A. A. Argyros, "Integrating tracking with fine object segmentation," *Image Vis. Comput.*, vol. 31, no. 10, pp. 771–785, 2013.

[21] S. Duffner and C. Garcia, "Fast pixelwise adaptive visual tracking of non-rigid objects," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2368–2380, May 2017.

[22] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

[23] S. C. Wong, V. Stamatescu, A. Gatt, D. Kearney, I. Lee, and M. D. McDonnell, "Track everything: Limiting prior knowledge in online multi-object recognition," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4669–4683, Oct. 2017.

[24] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2015, pp. 33–40.

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

[26] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 21–26.

[27] E. Gundogdu and A. A. Alatan, "Good features to correlate for visual tracking," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2526–2540, May 2018.

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

[29] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 1243–1248.

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

[31] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, "Fast optical flow using dense inverse search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 471–488.

[32] 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.

[33] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 564–586.

[34] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

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

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

[37] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1430–1438.

[38] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4293–4302.