Visual Tracking Under Motion Blur

Bo Ma, *Member, IEEE*, Lianghua Huang, Jianbing Shen, *Senior Member, IEEE*, Ling Shao, *Senior Member, IEEE*, Ming-Hsuan Yang, *Senior Member, IEEE*, and Fatih Porikli, *Fellow, IEEE*

***Abstract*— Most existing tracking algorithms do not explicitly consider the motion blur contained in video sequences, which degrades their performance in real-world applications where motion blur often occurs. In this paper, we propose to solve the motion blur problem in visual tracking in a unified framework. Specifically, a joint blur state estimation and multi-task reverse sparse learning framework are presented, where the closed-form solution of blur kernel and sparse code matrix is obtained simultaneously. The reverse process considers the blurry candidates as dictionary elements, and sparsely represents blurred templates with the candidates. By utilizing the information contained in the sparse code matrix, an efficient likelihood model is further developed, which quickly excludes irrelevant candidates and narrows the particle scale down. Experimental results on the challenging benchmarks show that our method performs well against the state-of-the-art trackers.**

***Index Terms*— Motion blur, tracking, sparse representation.**

I. Introduction

VISUAL tracking plays a critical role in computer vision with numerous applications such as surveillance, robotics and behavior analysis [1], [4], [35], [40], [41], [45], [50], [52]. Despite decades of studies, it is still a challenging task due to several complication factors in real world videos, e.g., background clutter, illumination variation, partial occlusions and object transformation. Tremendous efforts have been focused on establishing robust appearance models to handle these difficulties [5]–[12], [46]. However, most existing tracking algorithms do not explicitly consider the motion blur contained in video sequences, which degrades their performance in real world applications where motion blur is often unavoidable. Many state-of-the-art trackers, which achieve promising performance on sharp sequences, may easily fail on blurry ones.

A natural solution for this problem is to first perform deblurring on the blurred sequence, and then apply tracking on the deblurred one. However, several problems arise from this method. The first issue would be the negative effects of the ringing artifacts contained in the deblurred images, which are generated by the deconvolution methods due to the Gibbs phenomenon. Such noise creates harmful fake features and makes tracking difficult. Second, the expensive computational cost of most deblurring algorithms [13], [14] makes tracking slower. Furthermore, this method always ignores the similarity between target images in successive frames. Thus, the algorithm could not fully exploit the information and the deblurring and tracking performance will be both degraded.

Different from the traditional *deblurring and tracking* methods, some works try to avoid the noise-causing and inefficient deblurring step before tracking. In [15], the authors have observed that efficient tracking can be performed by directly matching blurred images instead of applying deblurring first. The blurred templates are obtained by performing convolutions on the clear images using blur kernels sampled under a Gaussian distribution. Based on this work, several notable studies [16]–[19] have been developed to improve performance of tracking under blur. Dai *et al.* [18] first estimate the direction of the target's motion blur using steerable filters, then traverse the blur strength l_b with a pixel step and find the best match for each l_b under the mean-shift algorithm, in which the blur strength is chosen with the highest score. In [16], a more standard model called directional blur is used to replace the translational Gaussian kernel. The observation is that the opening/closing operation of the shutter happens instantaneously, hence there is no Gaussian temporal blur. In [17], motivated by the success of sparse representation applied to vision tasks [20]–[24], [43], [47], [49], a unified sparse approximation framework is presented for integrating the visual tracking with the motion blur problem. The dictionary for sparse representation contains *normal templates, blur templates* and *trivial templates*. *Blur templates* are obtained by convolving the target image in the first frame with 64 blur kernels, which are obtained by sampling 8 different directions and 8 different speeds. The best candidate is chosen with the minimum reconstruction error.

The above mentioned methods basically approximate the target's blur state by sampling kernels. However, the sampled blur kernels could not accurately reflect the real blur states of the target, and they might fail when the degree of blur

Manuscript received January 26, 2016; revised July 14, 2016 and September 21, 2016; accepted October 3, 2016. Date of publication October 6, 2016; date of current version October 25, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61472036 and Grant 61272359, in part by the National Basic Research Program of China (973 Program) under Grant 2013CB328805, in part by the Australian Research Council's Discovery Projects Funding Scheme under Grant DP150104645, and in part by the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivana Tosic. (Corresponding author: Jianbing Shen.)

B. Ma, L. Huang, and J. Shen are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: bma000@bit.edu.cn; huanglianghua@bit.edu.cn; shenjianbing@bit.edu.cn).

L. Shao is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K. (e-mail: ling.shao@ieee.org).

M.-H. Yang is with the School of Engineering, University of California at Merced, Merced, CA 95344 USA (e-mail: mhyang@ucmerced.edu).

F. Porikli is with the Research School of Engineering, Australian National University, and NICTA (e-mail: fatih.porikli@anu.edu.au).

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

Digital Object Identifier 10.1109/TIP.2016.2615812

is beyond their representation scope. In addition, this kind of methods often generates highly redundant blur templates, causing repeatedly useless matching and extra computational burden. In this work, we attempt to tackle the above issues. On the one hand, we hope that the blur kernel can be explicitly estimated according to the target's real blur state instead of being approximated by sampling, so that the blurry candidates can be represented by templates more accurately. On the other hand, we want to avoid the noise/artifacts caused by deblurring, which will decrease the tracking performance. Furthermore, we expect that the blur kernel estimation and visual tracking should be jointly conducted instead of independently performed. In this way, the blur kernel would be more precisely estimated due to the consideration of correlations between candidates and templates, and in turn, tracking would be more robustly performed thanks to the correctly estimated blur kernel.

Motivated by the above ideas, we propose to accomplish tracking under motion blur in a joint blur kernel estimation and multi-task reverse sparse learning model. The blur kernel k and the sparse coding matrix C are obtained simultaneously within one optimization procedure. To avoid introducing deblurring noise, the estimated kernel k is not used for restoring candidates but for convolving with the templates to get the blurred templates. The reverse process indicates that the algorithm considers blurry candidates as dictionary atoms and the blurred templates as observations, and blurred templates are sparsely represented by blurry candidates. Since the number of templates is much smaller than that of candidates, the implementation will be more efficient. As all the sharp templates share one blur kernel for accurately representing the current target image, instead of solving the sparse learning problem for each template independently, we propose to solve the joint model in a multi-task manner.

After the blur kernel and sparse representation are computed, we need to find a robust and efficient way of particle selection to locate the target accurately. Different sparse representation based methods construct appearance models in various ways. Basically, they can be categorized into holistic sparse representation based models and local sparse feature based models. For example, in [23], [24], and [31], the candidates are evaluated by using reconstruction errors on a learned dictionary, whereas in [33], [34], and [36], the evaluation is performed directly on the local sparse codes. Multi-Task Tracking (MTT) tracker [27] also uses multi-task sparse representation for the construction of observation models. However, the differences between our method and MTT tracker are significant. First, the MTT tracker focuses on the l1 tracker [23] with faster and more accurate implementation. It is a general tracker which aims at handling challenges in normal videos. In contrast, our method is designed to specially tackle motion blur in tracking, and it aims at jointly estimating the blur kernel and performing tracking with the help of the multi-task sparse representation framework. Second, the MTT tracker uses reconstruction errors for likelihood evaluation, while ours uses the max pooling of sparse code matrix for the initial candidate screening, then we evaluate the rest candidates with a structured evaluation scheme. Third, we use reverse representation, which takes candidates as dictionary to represent blurred templates to accelerate the implementation. Finally, we do not require trivial templates since our multi-task reverse sparse representation model is mainly applied for blur kernel estimation instead of noise suppression.

In this work, we propose a two-stage scheme for effectively and efficiently filtering the candidates. In the first stage, where a holistic model is used, we perform a fast rejection scheme based on the coding matrix C to quickly narrow the particle scope down. We observe that the coefficients of biased candidates are either zeros or very small, so an evaluating scheme based on the values of sparse codes would quickly exclude most candidates. In the second stage, the very few survivors are further evaluated with a robust local sparse coding model. Candidates are separated into several parts, and evaluated block-wisely with the structured reconstruction errors. The overview of our tracking framework is illustrated in Fig. 1. Our source code will be available at.[1]

Compared to the existing approaches, the proposed visual tracking method offers the following contributions:

- To the best of our knowledge, we are the first to combine blur kernel estimation and visual object tracking in a unified framework, which jointly optimizes for the blur kernel and the sparse representation.
- We propose an iterative optimization algorithm for the multi-task model, which simultaneously obtains multiple sparse coding results and a single blur kernel of the candidates.
- Based on the insight on the sparse code matrix, we propose an efficient likelihood model to quickly exclude most irrelevant candidates for efficient visual tracking.

II. Joint Blur Kernel Estimation and Multi-Task Reverse Sparse Learning

In this section, we present the unified framework which combines the multi-task reverse sparse representation and blur kernel estimation in detail. We first discuss the original joint model and the motivation of this work. Next, we describe the proposed Multi-Task Reverse Sparse Representation (MTRSR) model. The optimization procedure is then introduced.

A. Problem Formulation

The proposed tracking method is implemented under the particle filter framework [25]. Denote y as one of the motion-blurred candidates in the current frame, given its estimated blur kernel k and white Gaussian noise ϵ, the blurry image y can be modeled as:

$$y = k * x + \epsilon, \tag{1}$$

where x is the latent sharp image of y and $*$ denotes the convolution operator. Deblurring the candidate y corresponds to the estimation of its latent image x and the blur kernel k:

$$\left\{\hat{x}, \hat{k}\right\} = \arg\min_{x,k} \|k * x - y\|_2^2 . \tag{2}$$

[1] http://github.com/shenjianbing/blurtracking

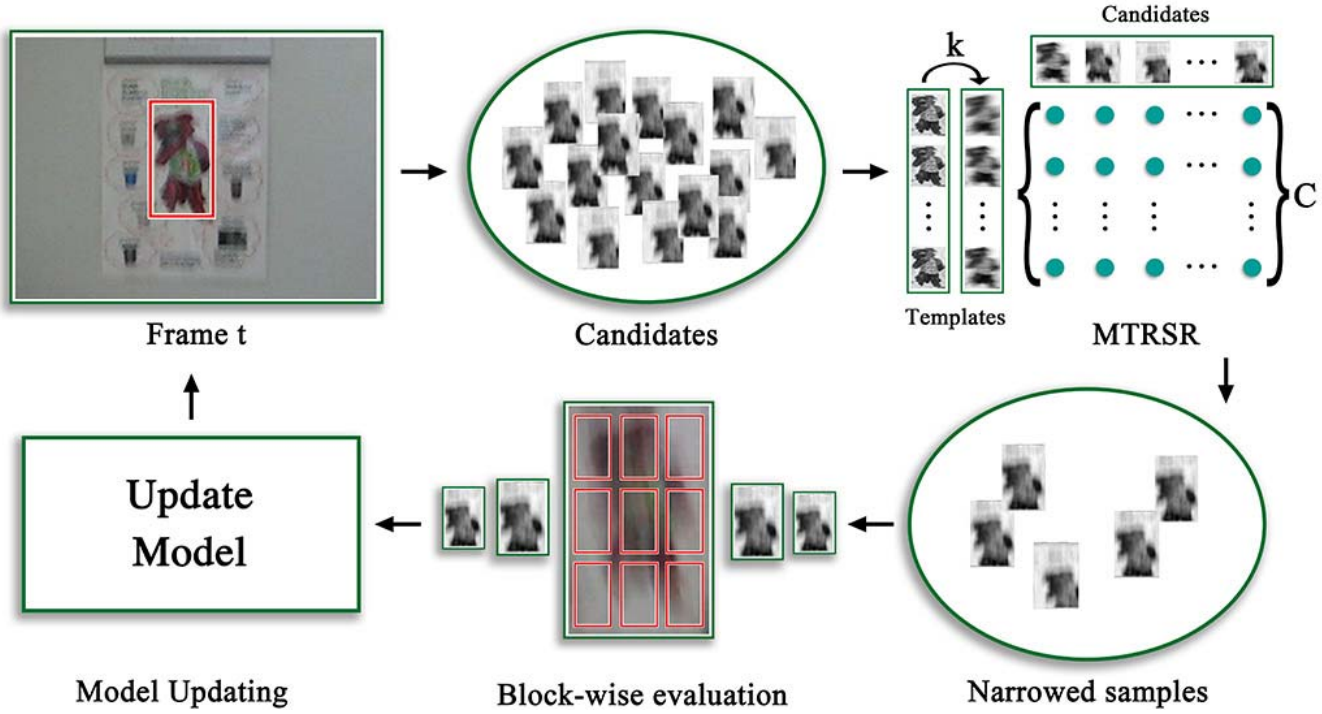

Fig. 1. The overall framework of the proposed visual tracking under motion blur algorithm.

This is an ill-posed inverse problem. We need to regularize x and k in (2) to obtain an accurate and stable solution,

$$\left\{\hat{x}, \hat{k}\right\} = \arg\min_{x,k} \|k * x - y\|_2^2 + \tau\rho\left(x\right) + \gamma \|k\|_2^2, \quad (3)$$

where $\rho\left(x\right)$ is a regularization term to make the final solution $\hat{x}$ smoother.

In sparse representation based tracking methods, if candidate y is close to the target, its deblurred image x should be well sparsely represented by the target's sharp template set T:

$$\hat{\alpha} = \arg\min_{\alpha} \|x - T\alpha\|_2^2 + \lambda\|\alpha\|_1, \quad (4)$$

where α is the sparse coefficient vector. By combining deblurring (3) and sparse representation (4) in a unified framework, we can get a joint model, which simultaneously deblurs the candidate y and computes its sparse representation on T [26]:

$$\left\{\hat{x}, \hat{k}, \hat{\alpha}\right\} = \arg\min_{x,k,\alpha} \|k * x - y\|_2^2 + \eta \|x - T\alpha\|_2^2 + \lambda\|\alpha\|_1 + \tau\rho\left(x\right) + \gamma \|k\|_2^2. \quad (5)$$

Using (5) directly for tracking in blurry sequences might be attractive, since it jointly deblurs the candidate and computes the sparse coefficients with the deblurred image, which seems to be more robust. Unfortunately, applying the model directly for tracking might actually be both inefficient and ineffective. Firstly, the ringing artifacts contained in the deblurred candidate images create deteriorated features and make the representation inaccurate and unstable, which further degrades the tracking performance. Moreover, the optimization process actually computes the sparse coefficients, estimates the blur kernel and performs deblurring for every single candidate, which makes the computing process very slow.

B. The Proposed MTRSR Model

To address the issues mentioned above, we do the following analysis. Note that deblurring the candidates is not an essential part in tracking tasks. To avoid the computationally expensive deblurring step, we can instead represent the blurry candidates with blurred templates as previous works do [15]–[17]. The blurred templates could be obtained by convolving the sharp templates with the estimated blur kernel k. By solving a sparse representation problem for each observation independently, the total computational cost is proportional to the number of candidates. We observe that the number of templates is far less than that of candidates. Therefore, similar to [2], if we consider the candidates as dictionary atoms and blurred templates as observations in turn, the computational cost of sparse coding will be significantly reduced [3].

Nonetheless, independently solving each sparse representation problem still raises some problems. In particular, the blur state of the target in one frame is unique, however, the above mentioned solving process estimates different blur kernels for different templates. This ignores the fact that different sharp templates share the same blur kernel for accurately representing the target. Furthermore, solving multiple independent sparse representation problems is still a time-consuming task.

According to the above analysis and motivated by the work in [27], we propose to build our joint blur kernel estimation and sparse representation model in a multi-task manner. We propose a MTRSR model, which combines multiple sparse representation problems in a joint model and is formulated as:

$$\left[\hat{k}, \hat{C}\right] = \arg\min_{k,C} \|k * T - YC\|_F^2 + \nu \|k\|_2^2 + \lambda_2\|C\|_{2,1}, \quad (6)$$

where k is the blur kernel, Y is the blurry candidate set used for representing blurred template set $k * T$, $*$ denotes the convolution operator and C is the sparse coefficient matrix.

Specifically, $\|C\|_{p,q} = \left[\sum_{i=1}^{N} \left(\|C_i\|_p\right)^q\right]^{1/q}$, where $\|C_i\|_p$ is the L_p norm of C_i, the i-th row of matrix C. Note that the regularization term $\rho(x)$ on the deblurred image x is dropped since we do not obtain the deblurred image x during optimization. In the MTRSR model formulated by (6), the sharp templates T are convolved with blur kernel k to get the blurred templates $k * T$, and are then sparsely represented by blurry candidates Y. Only one kernel k is estimated instead of multiple ks for different templates, which makes the solution more accurate and stable. The blurred templates convolved with k could represent the good candidates more precisely, and the sparse representation can also optimize the solution space of blur kernel k in turn.

C. Optimization

The MTRSR model contains two variables - we separate the optimization into two sub-problems and adopt the alternating minimization scheme to iteratively optimize the two variables. We first initialize the sparse coding matrix C by solving

$$\hat{C} = \arg\min_{C} \|T - YC\|_F^2 + \lambda_2 \|C\|_{2,1}. \tag{7}$$

where (7) is a multi-task sparse learning problem which can be solved by the Accelerated Proximal Gradient method [28].

1) Subproblem A (Optimizing k): With a fixed sparse coding matrix C, the blur kernel k can be estimated by solving the following optimization problem:

$$\hat{k} = \arg\min_{k} \left\| k * T - Y\hat{C} \right\|_F^2 + \nu \|k\|_2^2, \tag{8}$$

where $\hat{k} \in \mathbb{R}^{w_T \times h_T}$ is the estimated kernel, w_T and h_T are the width and height of a template, $\|k\|_2^2$ is a regularization term for suppressing most entries in k to reduce the boundary effects.

The minimization is a least squares problem with Tikhonov regularization. It has a closed-form solution [26]:

$$\hat{k} = F^{-1}\left(\frac{\bar{F}(T) \otimes F\left(Y\hat{C}\right)}{\bar{F}(T) \otimes F(T) + \nu I}\right), \tag{9}$$

where $F(\cdot)$ denotes Fast Fourier Transform, $F^{-1}(\cdot)$ denotes inverse Fast Fourier Transform, $\bar{F}(\cdot)$ denotes the complex conjugate of $F(\cdot)$, $\otimes$ denotes element-wise multiplication, and I is an identity matrix.

2) Subproblem B (Optimizing C): Given the estimated blur kernel k, the objective function can be rewritten as

$$\hat{C} = \arg\min_{C} \left\| \hat{k} * T - YC \right\|_F^2 + \lambda_2 \|C\|_{2,1}, \tag{10}$$

This is a multi-task sparse learning problem and can be readily solved by the Accelerated Proximal Gradient method [28]. The overall optimization procedure of the tracking model is summarized in Algorithm 1. The optimization is very efficient, since k has a closed-form solution and multiple sparse codings are solved within one model in a multi-task fashion. The algorithm converges in about 6-10 iterations in our experiments.

Algorithm 1 Optimization Algorithm of the Tracking Model

Input: template set T, candidate set Y, parameters ν and λ_2 and maximum number of iterations n.
Output: k and C.
1: Initialization: C is obtained by (7).
2: t=1.
3: **while** $t < n$ **do**
4: optimizing k: Solving k with fixed C using (9).
5: optimizing C: Solving C with fixed k by (10).
6: $t = t + 1$.
7: **end while**

III. Likelihood Model

The proposed tracking model is based on the particle filter framework [25]. The likelihood model in the framework is described in this section.

A. Fast Rejection of Irrelevant Candidates

The distribution of nonzero elements in matrix C indicates the similarity between the candidate set and the template set. In the reverse representation manner, templates tend to be sparsely represented by good candidates, while candidates that are too different from the target usually correspond to all zero coefficients. Based on the observation, we propose a strategy to efficiently exclude irrelevant candidates. Supposing n templates $T = T_1, T_2, \cdots, T_n$ are sparsely represented by m candidates $Y = \{Y_1, Y_2, \cdots, Y_m\}$, the sparse coding matrix $C = [\alpha_1, \alpha_2, \cdots, \alpha_n] \in \mathbf{R}^{m \times n}$, where α_i is the sparse coefficient vector of template T_i. Candidate Y_j is chosen for further evaluation only if

$$\max\left\{\alpha_1^j, \alpha_2^j, \cdots, \alpha_n^j\right\} > 0, \tag{11}$$

where α_1^j is the j-th element in α. Candidates that fail to match the condition are considered as irrelevant candidates and are rejected from further evaluation. After the rejection process, the rest of the candidates are narrowed down to a smaller set $Y^* = \{Y_1^*, Y_2^*, \cdots, Y_p^*\}$, and $p \ll m$ since C is highly sparse.

B. Structural Evaluation

The scope of the candidate set is largely narrowed down, which allows us to employ more time-consuming but accurate evaluating methods. Considering that, compared to a holistic model, a local model is more robust in handling local noise, partial occlusions and target transformation, we apply the structural reconstruction errors to evaluate the likelihoods of candidates. Each blurred template in T^* is separated into N overlapping patches. In this way, we can get Nn patches, and these patches are used for constructing a dictionary $D = [d_1^{(1)}, \cdots, d_N^{(1)}, \cdots, d_1^{(i)}, \cdots, d_N^{(i)}, \cdots, d_1^{(n)}, \cdots, d_N^{(n)}] \in \mathbf{R}^{d \times (Nn)}$. Each candidate Y_j is separated into patches

$\{y_k|k = 1, \cdots, N\}$ the same way as templates do. Each y_i is encoded by dictionary D:

$$\min_{\beta_k} \|y_k - D\beta_k\|_2^2 + \lambda_3\|\beta_k\|_1, \tag{12}$$

where $\beta_k \in \mathbf{R}^{(Nn)\times 1}$ indicates the sparse coefficients of y_k. If candidate Y_j is close to the target, its local patch y_k should be well represented by the corresponding sub-dictionary $D_k = [d_k^{(1)}, d_k^{(2)}, \cdots, d_k^{(n)}] \in \mathbf{R}^{d\times n}$. The corresponding sub-coefficients are $\beta_k^* = [\beta_k^k, \beta_k^{N+k}, \cdots, \beta_k^{(n-1)N+k}] \in \mathbf{R}^{n\times 1}$ where β_k^j is the j-th element of β_k. The corresponding reconstruction error for patch y_k is

$$\varepsilon_k = \|y_k - D_k\beta_k^*\|. \tag{13}$$

After the reconstruction errors of all the patches $\varepsilon_1 - \varepsilon_N$ are computed, the likelihood model of candidate Y_j is constructed by

$$P \propto \sum_{k=1}^{N} \exp(-\omega\varepsilon_k). \tag{14}$$

where ω denotes the scaling factor.

IV. Update Scheme

To adapt to the target's appearance variation, templates need to be updated overtime. In tracking under motion blur, a common idea of model updating is to deblur the estimated candidate and add it into the template set. However, the noise contained in the deblurred images could deteriorate the templates. As motion blur is generally temporarily appeared in most cases, in our update scheme, we only consider those tracking results whose images are relatively sharp.

We obtain the convolved template set T^* by convolving the sharp template set T with the estimated blur kernel k in the current frame. The dissimilarity value δ between sets T^* and T is calculated as

$$\delta = \frac{1}{n}\sum_{i=1}^{n} \left\|T_i^* - T_i\right\|_2^2, \tag{15}$$

where T_i^* and T_i are the i-th templates of T^* and T respectively. It is obvious that T^* is close to T if δ is very small. When $\delta < \delta_0$ which is a predefined dissimilarity threshold, we deem the tracking result blur free, and replace the i-th template in T with the tracking result. i is chosen by the following criterion:

$$i = \arg\max_k \left\{ \left\|T_k^* - T_k\right\|_2^2 \mid k = 1, \cdots, n \right\}. \tag{16}$$

V. Experimental Results

Our method is implemented in MATLAB R2012a and runs at 11.7 fps on an Intel Core i5 2.5GHz CPU with 4G memory. We maintain 10 templates during tracking and sample 600 candidates in each frame, all of them are normalized to 32×32. When performing the structured evaluation, 9 overlapped local patches (16×16) are extracted within each frame with 8 pixel as the step length. λ_2, λ_3 and γ are fixed to 0.01, π is set to 5, ν is set to 0.01 and δ_0 is set to 0.03 in all experiments. In order to comprehensively evaluate our approach, we first evaluate our tracker on 58 sequences where objects are under severe motion blur. Then we present our results on a general benchmark [4] to demonstrate that our tracker can also perform well on non-blurry sequences.

A. Performance on Blurry Sequences

In this section, we present the experimental results by our method on 58 blurry sequences where objects are under severe motion blur. 51 of the sequences are obtained by convolving the sharp videos in benchmark [4] with randomly sampled blur kernels, and the rest sequences (i.e., BlurBody, BlurCar1, BlurCar2, BlurCar3, BlurCar4, BlurFace and BlurOwl) are acquired from the blurry videos in OTB-100 [48]. Our approach is compared with 13 recent state-of-the-art tracking methods including Multi-Task Tracking (MTT) tracker [27], Kernelized Correlation Filters (KCF) [44], Discriminative Scale Space Tracker (DSST) [51], Struck [10], Color-attribute based tracker (CNT) [42], Circulant Structure tracker with Kernels (CSK) [38], Sparsity-based Collaborative Model (SCM) [32], BLUr-driven Tracker (BLUT) [17], Compressive Tracker (CT) [30], Adaptive correlation filters based tracking (MOSSE) [37], Least Soft-threshold Squared Tracker (LSST) [31], Spatio-Temporal Context tracker (STC) [39] and Adaptive Structural Local Apperance model (ASLA) [29], where ASLA, MTT and SCM are sparse representation based methods and BLUT is a blur-driven object tracker. These trackers are evaluated using the source codes from the original authors and each is run with carefully tuned parameters. Since fast motion of objects are common in blurry sequences, we set larger search radius for the trackers to cover possible target locations.

1) Overall Performance: The precision plots and success plots [4] are applied to evaluate the overall performance of our algorithm and compared trackers. The precision plots indicate the percentage of frames whose estimated location is within the given threshold distance to the ground truth. The success plots demonstrate the ratios of successful frames whose overlap rate is larger than the given threshold. The precision score is given by the score on a selected representative threshold (e.g., 20 pixels). The success score is evaluated by the area under curve (AUC) of each tracker. Fig. 4 shows the precision plots and success plots of the trackers on 58 blurry videos. For precision plots, we rank the trackers according to the results at the error threshold of 20 pixels. For success plots, the trackers are ranked according to the AUC scores.

The precision scores and AUC scores for each tracker are shown in the legend of Fig. 4. From Fig. 4, we can see that KCF, DSST and our tracker perform well on the 58 blurry sequences, and our tracker achieves the best performance. Both the KCF and the DSST trackers belong to the correlation filter based tracking methods. Their stable tracking results may be attributed to the advantage of the correlation filter in handling blurry images. The discriminative ability of the HOG feature also contributes to the performance of KCF. In the precision plots, our algorithm performs 0.9% better than KCF

Fig. 2. Representative results of different trackers on sequences *Car1*, *Owl* and *Car3*. Objects in these sequences are under heavy motion blur.

and 5.7% better than DSST. In the success plots, our tracker outperforms KCF by 2.3% and DSST by 1.4.

It is also observed that our tracker significantly outperforms the blur-driven tracker BLUT. It is mainly because, compared to BLUT, which uses 64 predefined blur kernels obtained offline to capture different blur effects, in our method, the blur kernel is estimated with the candidates and updated online to adaptively reflect the blur state of the target, which is more accurate and computationally efficient. Besides, comparison between our tracker and the related MTT tracker indicates a significant improvement (64.9% versus 36.9%). The results suggest the contribution of the blur kernel estimation in improving the performance of our approach.

Overall, our tracker performs excellently on these blurry sequences compared to other trackers. The leading causes are summarized as follows. First, the estimated and online updated blur kernel in the tracking model truly reflects the blur state of the target, which makes the algorithm more robust against motion blur. Second, the multi-task reverse sparse representation, which considers the correlation among templates, greatly improves both computational efficiency and tracking performance. Third, the structured representation in the likelihood model further improves the robustness of our method against local noise and partial occlusions.

2) Qualitative Evaluation: A qualitative evaluation of our algorithm is presented in this section. For the 58 blurry videos, extreme motion blur is the main challenge for visual tracking. Additionally, there are some other challenges such as illumination variation, partial occlusion and in-/out-of plane rotation. We select 12 representative videos from them and discuss the tracking performance of different trackers as follows.

a) Motion blur: As shown in Fig. 2, the targets in sequences *Car1*, *Car4* and *Owl* are under significant motion blur in some frames caused by fast motion of cameras. In sequence *Car1*, the camera shakes throughout the sequence and the car in the video is severely blurred in several frames. Struck and SCM easily drift away when the target is not sharp, and BLUT and LSST also lose the target when the car is largely blurred. ASLA, KCF and our tracker achieve the best performance and our tracker obtains the most accurate results. In sequence *Car3*, which is similar to *Car1* with a shaking camera, but the motion blur is less severe. Most trackers lock the target well, but Struck and KCF sometimes shift several pixels away. Our tracker obtains stable tracking results. In sequence *Owl*, the camera shakes strenuously and the target is under severe motion blur. ASLA, KCF and L1APG sometimes mistakenly track the target when fast motion blur appears. BLUT, Struck and our tracker perform better in this sequence.

b) Motion blur + illumination variation: Car4 and *Singer1* in Fig. 3 are two sequences where targets are under motion blur as well as illumination variation. In sequence *Car4*, the car drives through a bridge and the light condition changes significantly. Motion blur is added manually by convolving with a random blur kernel. We can observe from Fig. 3 that KCF and L1APG fail to track the target. SCM and Struck drift away when motion blur occurs. The ASLA tracker significantly mistakenly estimates the scale. Only BLUT, LSST and our tracker successfully lock the target throughout the tracking process. In sequence *Singer1*, besides manually performed motion blur and illumination variation, the scale of the target is also gradually changed. Most trackers (e.g., ASLA, KCF, LSST) mistakenly estimate the scale or drift away from the location of the singer. BLUT, SCM and our tracker achieve the best performance. The blur-driven tracking model in BLUT makes it robust in predicting the location of the blurred target, and the normalized local intensity features make SCM less vulnerable to illumination changes. The accuracy of our method could be attributed to the structured sparse representation in handling local noise and light condition change and the estimated blur kernel to deal with blurred targets.

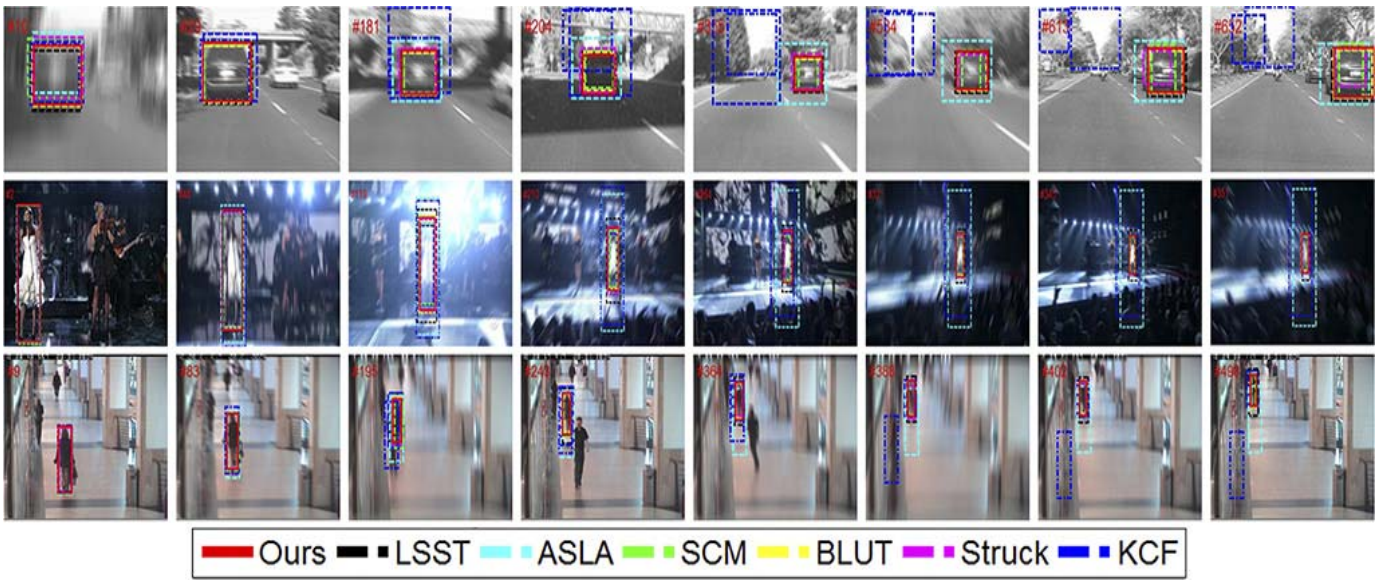

Fig. 3. Representative results of different trackers on sequences *Car4*, *Singer1* and *Walking2*. Objects in these sequences are under heavy motion blur. Besides, in sequences *Car4* and *Singer1*, targets are under significant illumination variation, and in sequence *Walking2*, the object suffers from partial occlusions.

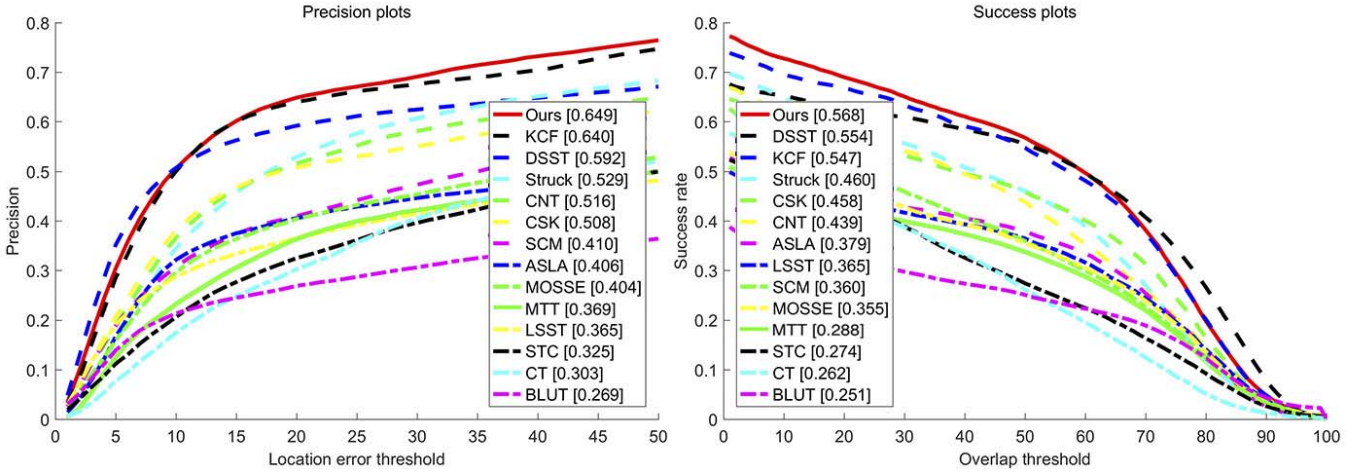

Fig. 4. Precision plots and success plots over 58 blurry video sequences. The legends in the left sub-figure and the right sub-figure show the precision scores and AUC scores for each tracker, respectively.

c) Motion blur + partial occlusion: Sequence *Walking2* is selected for testing trackers' robustness against occlusions under motion blur. A woman walks through a corridor and is occluded by a man in some frames. Motion blur is performed manually with random blur kernels. KCF and ASLA lose the target as occlusion appears (e.g., #216, #269). L1APG and LSST drift several pixels away in the blurry frames. BLUT and our method achieve the most accurate results. The robustness of our tracker against occlusion could be attributed to the structured representation that applies a blocking scheme to separate the target into several overlapped blocks - in this way, the adverse effect of partial noise would be alleviated. The estimated and online updated blur kernel also helps our method handle the motion blur problem in distinguishing the occluded target.

d) Motion blur + fast moving: Fig. 5 demonstrates the tracking results in three sequences (i.e., *Deer*, *Face* and *Jumping*) with fast motion. In sequence *Deer*, the deer runs fast and the target is tarnished when severe motion blur appears in some frames. L1APG, Struck, ASLA and BLUT fail to locate the target at frames with blurry object images. Our tracker locks the head of the deer throughout the sequence. BLUT also performs relatively well in this sequence. In sequence *Face* the camera moves fast and the object's motion blur is severe. Also, the target slightly rotates in some frames. Struck, KCF and ASLA sometimes drift away when the target is not so sharp, as these methods do not explicitly consider motion blur in tracking sequences. In the *Jumping* sequence, the motion of the tracking target is so drastic that ASLA and L1APG fail before frame #41. LSST, BLUT and our method can keep track of the target to the end, but our method achieves more accurate tracking results. The estimated blur kernel and effectiveness of multi-task sparse representation make our method sail through the fast motion sequences.

e) Motion blur + in-/out-of-plane rotation: To evaluate our method in more general cases, we selected some sequences

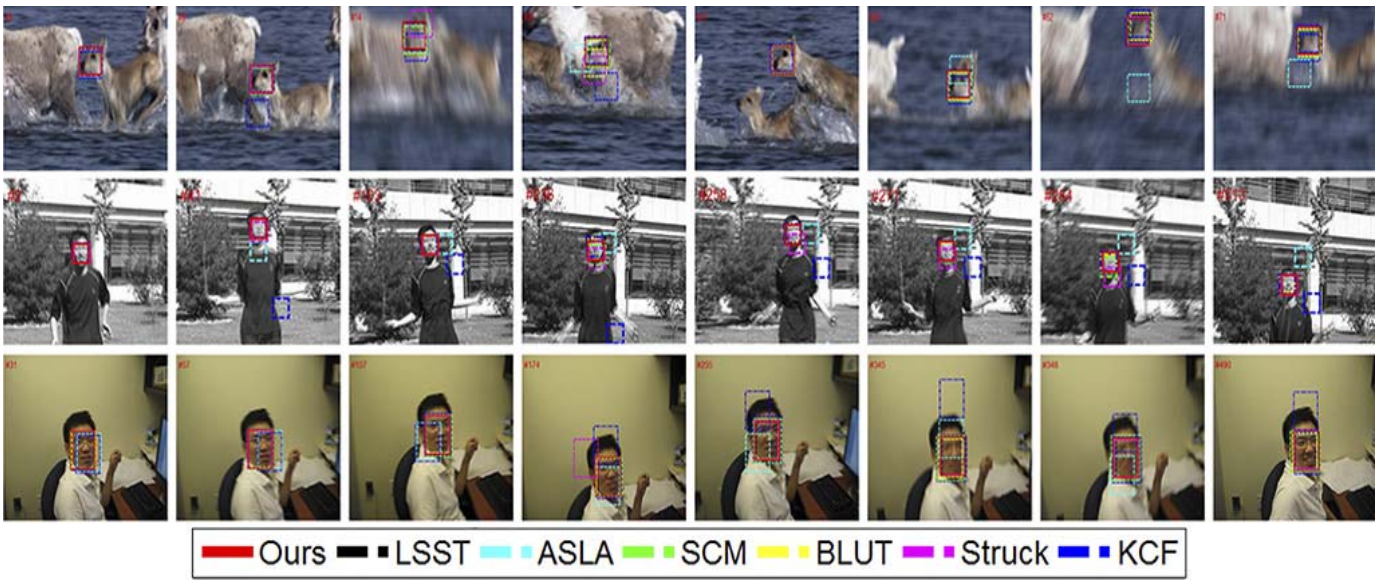

Fig. 5. Representative results of different trackers on sequences *Deer*, *Jumping* and *Face*. Fast moving of objects is the main challenge in these videos.

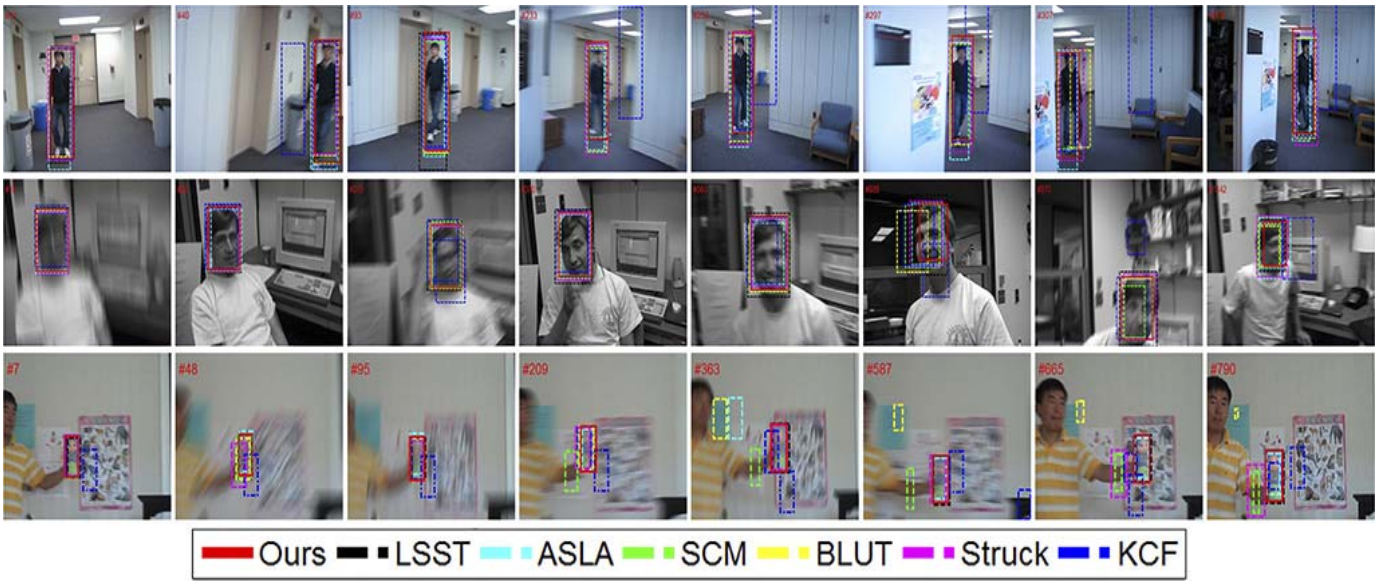

Fig. 6. Representative results of different trackers on sequences *Body*, *Dudek* and *Doll*. Besides motion blur, in-plane/out-of-plane rotation is the additional challenge throughout these sequences.

(i.e., *Body, Dudek* and *Doll*) where in-plane or/and out-of-plane rotations are additional challenges along with motion blur. It can be observed from Fig. 6 that rotation of the target makes it much more indistinguishable in a new frame and casts a more difficult problem in tracking. In sequence *Body*, L1APG, ASLA, BLUT and Struck drift several pixels away or fail to estimate the correct scale. SCM and our method lock the target with accurate scale estimation, and our method achieves more stable results. In sequence *Dudek*, the person rotates his head for about 360 degrees. L1APG, LSST, BLUT and KCF sometimes drift away when drastic motion blur occurs (e.g., #613, #668, #962). Only our method locks the target till the end with the correct scale. In sequence *Doll*, BLUT, ASLA, SCM, KCF and L1APG lose the target (e.g., in frames #198, #323 and #425). Only LSST and our tracker successfully track the doll in the whole sequence.

B. Performance on Benchmark

To evaluate the overall performance that our tracker can also perform well on non-blurry sequences, we carried out experiments on the complete benchmark [4], which contains 51 sequences with various challenging factors such as partial occlusions, object deformation, fast motion, illumination change and scale variation. We compare our results with all others recommended in [4], such as Struck [10], Sparsity-based Collaborative Model [32], Tracking Learning Detection (TLD) [11], Adaptive Structural Local Appearance tracker (ASLA) [29], Compressive Tracker (CT) [30], L1 tracker using Accelerated Proximal Gradient

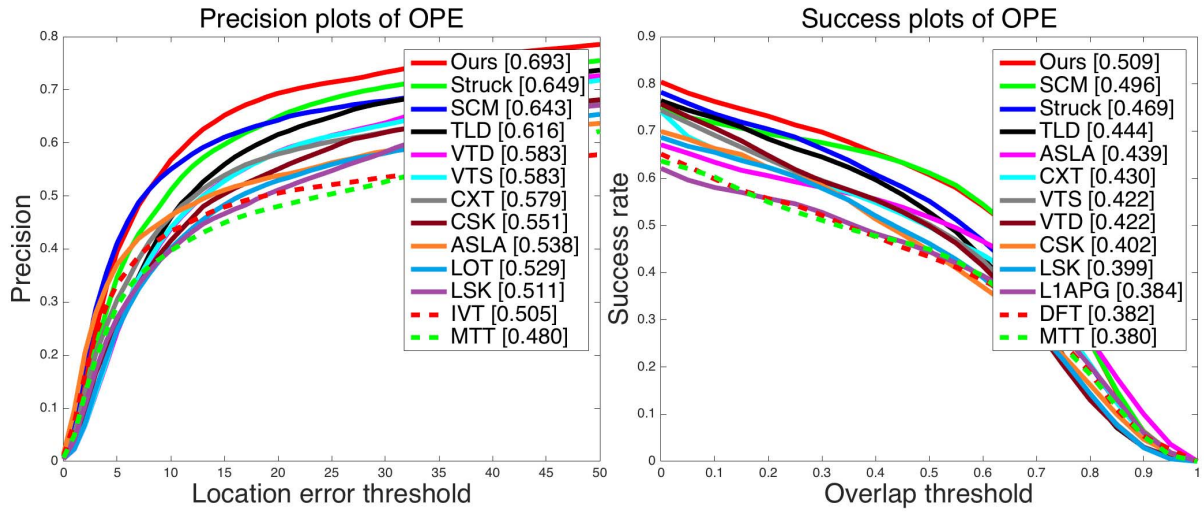

Fig. 7. Overall performance of our tracker against all others in the complete benchmark [4]. Only top 12 trackers and the MTT tracker are displayed.

approach (L1APG) [24], Least Soft-threshold Squares Tracker (LSST) [31] and Visual Tracking Detection (VTD) [12]. Most parameter settings remain the same as before. The only exception is that the ν, which is used for regularizing the blur kernel k, is set to 0.1 instead of 0.01, since the motion blur for most sequences are less serious than the 58 extremely blurry videos.

As shown in Fig. 7, our tracker achieves the best performance in terms of both the precision score and the success score. The competitive performance on the general benchmark indicates the overall robustness of our method. Though our approach is designed for tracking objects under severe motion blur, it can also be used on blur-free videos where general challenges such as occlusions, object transformation and background clutter exist. The fast rejection scheme and the block-wise evaluation in the likelihood model are effective for both types of sequences.

VI. Conclusion

To handle the motion blur during tracking, we have proposed a tracking model which integrates the blur kernel estimation and the sparse coding matrix calculation in a unified framework based on multi-task reverse sparse representation. The estimated blur kernel is applied to the normal templates to get the convolved templates which reflect the real blur situation of the target. The sparse coding matrix containing some useful information for distinguishing the target is used to select some better candidates. Then, we have constructed an effective likelihood model based on the structural reconstruction error to determine the best candidate. Comprehensive experimental comparisons with the state-of-the-art algorithms demonstrate the effectiveness of the proposed tracking method in dealing with motion blur.

References

[1] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[2] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[3] B. Zhuang, H. Lu, Z. Xiao, and D. Wang, "Visual tracking via discriminative sparse similarity map," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1872–1881, Apr. 2014.

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

[5] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2427–2434.

[6] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2011, pp. 1323–1330.

[7] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[8] M. M. N. Ali, M. Abdullah-Al-Wadud, and S.-L. Lee, "Multiple object tracking with partial occlusion handling using salient feature points," *Inf. Sci.*, vol. 278, pp. 448–465, Sep. 2014.

[9] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1910–1917.

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

[11] 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.

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

[13] A. Levin, "Blind motion deblurring using image statistics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 841–848.

[14] S. Dai and Y. Wu, "Motion from blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[15] H. Jin, P. Favaro, and R. Cipolla, "Visual tracking in the presence of motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 18–25.

[16] C. Mei and I. Reid, "Modeling and generating complex motion blur for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[17] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng, "Blurred target tracking by blur-driven tracker," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1100–1107.

[18] S. Dai, M. Yang, Y. Wu, and A. K. Katsaggelos, "Tracking motion-blurred targets in video," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 2389–2392.

[19] Y. Wu, J. Hu, F. Li, E. Cheng, J. Yu, and H. Ling, "Kernel-based motion-blurred target tracking," in *Proc. Adv. Vis. Comput.*, 2011, pp. 486–495.

[20] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.

[21] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 113–127.

[22] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.

[23] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.

[24] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1830–1837.

[25] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods Pract*, Springer, 2013.

[26] H. Zhang, J. Yang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang, "Close the loop: Joint blind image restoration and recognition with sparse representation prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 770–777.

[27] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2042–2049.

[28] X. Chen, W. Pan, J. T. Kwok, and J. G. Carbonell, "Accelerated gradient method for multi-task sparse learning problem," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 746–751.

[29] 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.*, Jun. 2012, pp. 1822–1829.

[30] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.

[31] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2371–2378.

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

[33] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Online discriminative object tracking with local sparse representation," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2012, pp. 425–432.

[34] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang, "Transferring visual prior for online object tracking," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3296–3305, Jul. 2012.

[35] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1940–1947.

[36] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1313–1320.

[37] 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.

[38] 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.

[39] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.

[40] B. Ma, H. Hu, J. Shen, Y. Zhang, and F. Porikli, "Linearization to nonlinear learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4400–4407.

[41] B. Ma, J. Shen, Y. Liu, H. Hu, L. Shao, and X. Li, "Visual tracking using strong classifier and structural local sparse descriptors," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1818–1828, Oct. 2015.

[42] 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.*, Jun. 2014, pp. 1090–1097.

[43] X. Lu, X. Li, and L. Mou, "Semi-supervised multitask learning for scene recognition," *IEEE Trans. Cybernetics*, vol. 45, no. 9, pp. 1967–1976, Sep. 2015.

[44] 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.

[45] B. Ma, L. Huang, J. Shen, and L. Shao, "Discriminative tracking using tensor pooling," *IEEE Trans. Cybernetics*, to be published, vol. 46, no. 11, pp. 2411–2422, Nov. 2016, doi: 10.1109/TCYB.2015.2477879.

[46] W. Wang, J. Shen, X. Li, and F. Porikli, "Robust video object cosegmentation," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3137–3148, Oct. 2015.

[47] B. Du *et al.*, "Exploring Representativeness and Informativeness for Active learning," *IEEE Trans. Cybernetics*, doi: 10.1109/TCYB.2015.2496974.

[48] 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.

[49] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybernetics*, vol. 46, no. 2, pp. 487–498, Feb. 2016.

[50] X. Dong, J. Shen, and L. Shao, "Sub-Markov random walk for image segmentation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 516–527, Feb. 2016.

[51] 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. 2014.

[52] B. Ma, H. Hu, J. Shen, Y. Liu, and L. Shao, "Generalized pooling for robust object tracking," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4199–4208, Sep. 2016.