Structure-Aware Local Sparse Coding for Visual Tracking

Yuankai Qi, Lei Qin, Jian Zhang, Shengping Zhang, Qingming Huang, *Fellow, IEEE*, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— Sparse coding has been applied to visual tracking and related vision problems with demonstrated success in recent years. Existing tracking methods based on local sparse coding sample patches from a target candidate and sparsely encode these using a dictionary consisting of patches sampled from target template images. The discriminative strength of existing methods based on local sparse coding is limited as spatial structure constraints among the template patches are not exploited. To address this problem, we propose a structure-aware local sparse coding algorithm, which encodes a target candidate using templates with both global and local sparsity constraints. For robust tracking, we show the local regions of a candidate region should be encoded only with the corresponding local regions of the target templates that are the most similar from the global view. Thus, a more precise and discriminative sparse representation is obtained to account for appearance changes. To alleviate the issues with tracking drifts, we design an effective template update scheme. Extensive experiments on challenging image sequences demonstrate the effectiveness of the proposed algorithm against numerous state-of-the-art methods.**

***Index Terms*— Visual tracking, local sparse coding, spatial structure information, template update.**

I. Introduction

VISUAL tracking has been one of the most fundamental topics in computer vision due to its key roles in numerous applications such as surveillance, human-computer interaction, automatic driving, and video analysis. It aims to estimate the states (*e.g.*, location, scale, rotation, *etc.*) of a target in a sequence of images after specifying the target in the first frame. While significant efforts have been made in the past decades, developing a robust tracking algorithm for complicated scenarios is still a challenging task due to factors like heavy occlusion, pose changes, large scale variations, camera motion, and illumination variations.

Broadly speaking, tracking methods can be categorized as either discriminative [1]–[7] or generative [8]–[17]. Discriminative methods learn a binary classifier to separate the target from its background and identify object locations with a local search. Much effort has been devoted to improving the discriminative power of classifiers. Grabner *et al*. [3] use online boosting to construct a feature selection algorithm and integrate several weak classifiers into a single strong one. A structured output support vector machine is employed by Hare *et al*. [7] to directly couple the classifier's output with the tracked target position. In [18], Babenko *et al*. introduce multiple instance learning into the training process to alleviate the confusion caused by multiple positive samples. Most recently, hierarchical features learned from convolutional neural networks are used to distinguish the tracked target from its background for object tracking [19], [20].

Generative approaches cast object tracking as a searching task to find the target candidate most similar to the tracked target. The key issues are concerned with how object appearance can be modeled and updated to account for large changes caused by occlusion, lighting variation, and view changes. In [14], incremental subspace learning is used to adapt to target appearance changes. Sparse coding has been applied to represent appearance for visual tracking [9], [21]–[24]. These sparse coding approaches assume that a good target candidate can be represented by a set of target templates with a small reconstruction error, and vice versa. According to the type of dictionary atoms used in sparse coding, these models can be classified into holistic sparse coding (HSC) [15], [21], [25], in which each dictionary atom is a whole target template image; and local sparse coding (LSC) [9], [24], in which the dictionary atoms are local patches sampled from target template images. As demonstrated in the benchmark study on object tracking [26], [27], LSC based trackers perform favorably among a total of 31 representative trackers.

However, the LSC scheme does not consider the spatial structural information among the used template patches, which should come from the templates that are the most similar to the currently considered candidate. The loss of spatial

Manuscript received September 14, 2016; revised April 3, 2017 and December 21, 2017; accepted January 9, 2018. Date of publication January 24, 2018; date of current version May 1, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61620106009, Grant 61332016, Grant U1636214, Grant 61650202, Grant 61572465, Grant 61390510, Grant 61732007, and Grant 61672188, in part by the Key Research Program of Frontier Sciences, CAS under Grant QYZDJ-SSW-SYS013, in part by NSF CAREER under Grant 1149783, and gifts from Adobe, Verisk, and Nvidia. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christopher Wyatt. *(Corresponding author: Qingming Huang.)*

Y. Qi is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: yk.qi@hit.edu.cn).

L. Qin is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qinlei@ict.ac.cn).

J. Zhang is with the Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia (e-mail: jian.zhang@kaust.edu.sa).

S. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China (e-mail: s.zhang@hit.edu.cn).

Q. Huang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and also with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: qmhuang@ucas.ac.cn).

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

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.2018.2797482

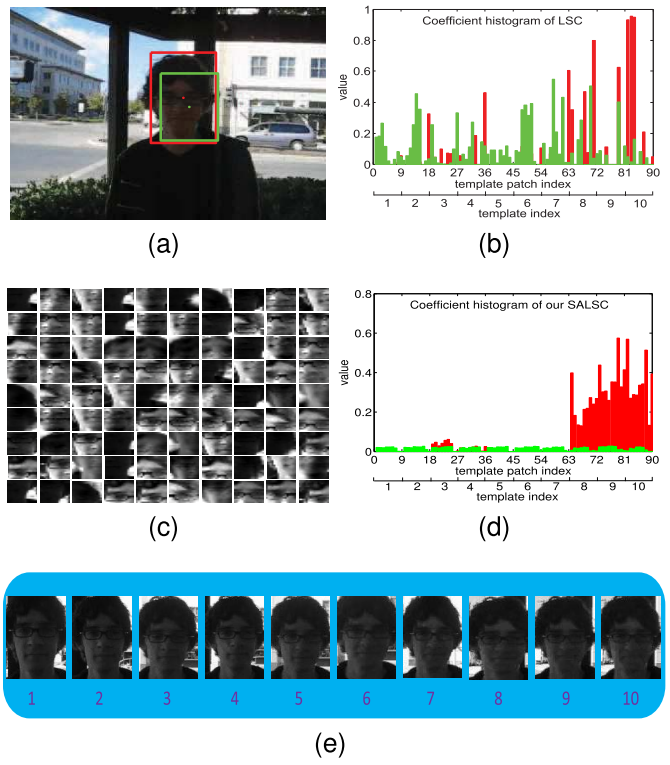


Fig. 1. Structure-aware local sparse coding generates a more discriminative sparse representation. The good (in the red box) and bad (in the green box) candidates in (a) are encoded with the dictionary (c), which consists of local patches extracted from templates (e). Please note that the templates are different (e.g., whether there is a pillar in the background or not.). The coefficient histograms obtained by LSC and SALSC are shown in (b) and (d).

structural information may result in the loss of the discriminative strength of the appearance model. Examples of two target candidates and their corresponding histograms of appearance representations using LSC are shown in Figure (1a) and Figure (1b), respectively. Since the histograms are similar, it is difficult to distinguish one from the other. A more intuitive example is shown in the left plane of Figure 2. In the upper left image, the region in the solid yellow box is the target to be tracked. In the upper right image, the regions in solid and dashed boxes are a good candidate and a bad candidate, respectively. Since the LSC methods represent each patch of a target candidate using patches from arbitrary templates, both the good and bad candidates can be well reconstructed with small errors as illustrated in the lower left image, and therefore tracking methods based on these representations cannot accurately locate the target in this scenario.

To address the aforementioned problems, Jia *et al*. [9] propose an alignment pooling method for object tracking. For a patch of the currently considered candidate, the alignment pooling counts only the coefficients corresponding to the template patches at the same position as the candidate patch. Zhong *et al*. [22] develop a collaborative model to encode a target candidate with both HSC as well as LSC, and combine the two scores with a multiplication operator for object tracking. While these approaches help alleviate the problems, the spatial structure information among the used template patches is not exploited.

In this paper, we propose the structure-aware local sparse coding (SALSC) method for object tracking. The SALSC scheme sparsely encodes the patches of a target candidate using a dictionary consisting of patches sampled from target template images. Unlike LSC, the SALSC method also exploits the spatial structure information with a global sparsity constraint on the coding coefficients, which minimizes the number of the templates used to encode the candidate and therefore implicitly chooses the templates that are more similar to the candidate. To this end, we design a rearranging function over the coefficients of patches and exploit the $\ell_{2,1}$ norm on the rearranged coefficients to constrain the global sparsity. As illustrated in the right image of Figure 2, the combination of the global and local sparsity constraints makes SALSC only use the patches of the templates that are the most similar to the currently considered target candidate. as shown in Figure 1b and Figure 1d. In addition, we present an algorithm to solve the posed optimization problem with the rearranging function.

For robust object tracking, an update scheme is often used. Although several effective update approaches have been proposed [3], [14], [21], it is not clear how old and new templates should be retained to properly model object appearance changes. Without an effective model update scheme, it is inevitable that trackers drift away from targets in complex scenes. In this work, we present an effective update strategy by exploiting long-term and short-term memory based on the recent studies of human memory [28]–[30]. Experimental results show that the proposed update scheme is effective in accounting for appearance changes for object tracking.

The main contributions of this work are summarized as follows:

1) We present the SALSC model to take both the global and local sparsity constraints into consideration for more effective appearance modeling for visual tracking.
2) We propose an optimization method to solve the proposed SALSC model and apply it to visual tracking with an effective template update scheme.
3) Extensive experimental results on a large tracking benchmark dataset demonstrate the effectiveness of the proposed tracking method compared to several state-of-the-art trackers.

II. Related Work

In this section, we briefly review the existing sparse coding based tracking methods. A more comprehensive review on sparse trackers can be found in [31]. In sparse coding based tracking methods, the image region of the tracked target is cropped as a *template*. Generally, numerous templates are collected to cover as many potential states of the target as possible. When the sparse dictionary is constructed based on patches locally sampled from all templates, it is referred to as a *local sparse coding* scheme. In contrast, when each dictionary atom is a whole template rather than a patch, it is referred to as a *holistic sparse coding* (HSC) scheme.

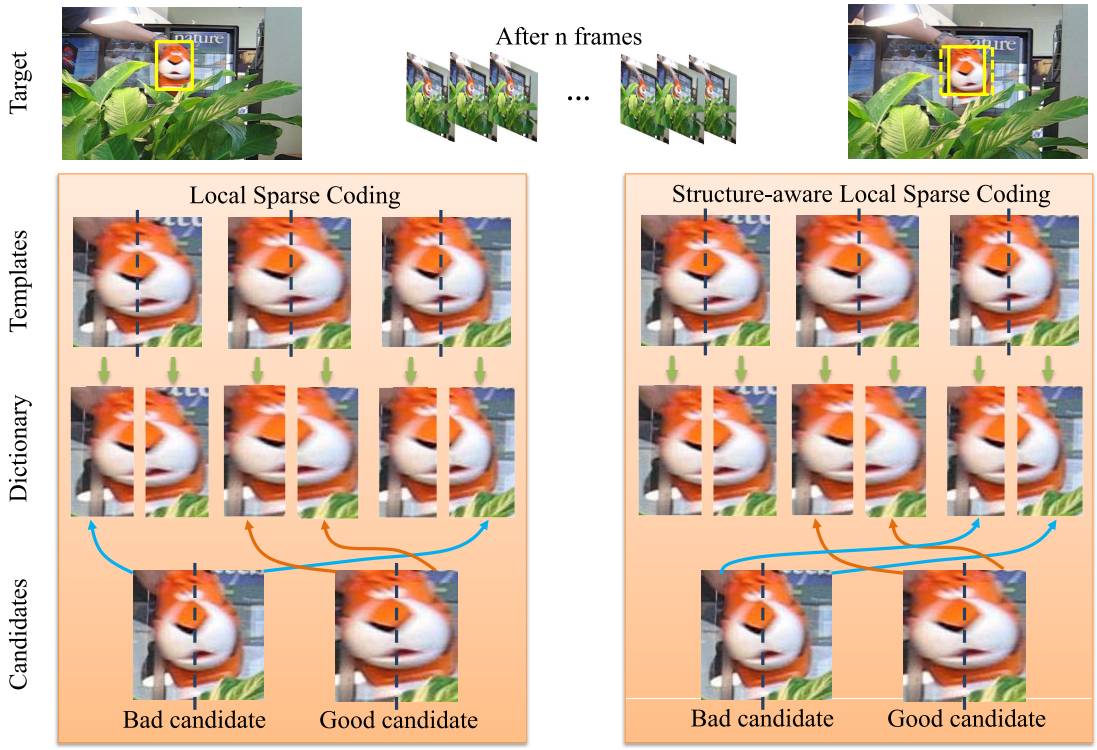


Fig. 2. LSC represents a target candidate using patches from *all* target templates. When the templates are not good enough (*e.g.*, some templates have more background information), the reconstruction error can no longer tell the good candidate from the bad one, since their reconstruction errors are very close. In contrast, the proposed SALSC represents a candidate using patches from *the candidate's most similar* templates. In this way, the good candidate can have a smaller reconstruction error than that of the bad candidate.

A. Holistic Sparse Coding Based Trackers

Mei and Ling [21] propose a HSC based tracking method which represents each target candidate using a set of target and trivial templates (a trivial template has only one non-zero element). The purpose of introducing trivial templates into sparse representation is to account for occlusions and noise. Let $\mathbf{x} \in \mathbb{R}^H$ denote a vectorized H-dimensional target candidate and $\mathbf{D}_h \in \mathbb{R}^{H\times(N+H)}$ the dictionary consisting of N vectorized templates (each template is H-dimensional) and H trivial templates. The sparse representation can be solved by

$$\begin{aligned}\mathbf{c}^* = \arg\min_{\mathbf{c}} \ & \|\mathbf{D}_h\mathbf{c} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{c}\|_1 \\ \text{s.t. } & c_i \geq 0 \quad \forall i = 1, 2, \ldots, N + H,\end{aligned} \tag{1}$$

where c_i denotes the i-th element of the coefficient vector $\mathbf{c}$, and $\|\cdot\|_1$ is the ℓ_1 norm that computes the sum of the absolute values of all elements in a vector. By minimizing both the reconstruction error (the first term) and global sparsity (the second term) in (1), the HSC scheme represents the candidate using as few target templates as possible that are the most similar to the tracked target. A good target candidate (*e.g.*, the one most similar to the target) can be reconstructed by the target templates and the associated coefficients. In contrast, in the case of a bad candidate, it cannot be well reconstructed by the target templates alone. Thus, the tracked target can be determined by finding the minimum reconstruction error using the target templates and the associated coefficients. While it has been shown to perform well in visual tracking, this method is less effective when the target undergoes self-variations, such as pose changes, as the trivial templates used by HSC are not flexible enough to handle such complicated variations.

B. Local Sparse Coding Based Trackers

In the LSC scheme, the dictionary is constructed based on patches locally sampled from all target templates [9]. Let $\mathbf{D}_l = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_{N\times M}] \in \mathbb{R}^{L\times(N\times M)}$ be the dictionary, where L is the dimension of each vectorized patch, and N and M are the number of templates and the number of patches sampled from each template, respectively. Let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M] \in \mathbb{R}^{L\times M}$ denote the patches sampled from a target candidate. A candidate is represented in a patch-wise manner by solving

$$\begin{aligned}\mathbf{C}^* = \arg\min_{\mathbf{C}} \ & \|\mathbf{D}_l\mathbf{C} - \mathbf{Y}\|_F^2 + \lambda\sum_{i=1}^{M}\|\mathbf{c}_i\|_1 \\ \text{s.t. } & c_{ij} \geq 0 \quad \forall i = 1, 2, \ldots, N \times M, \ \forall j = 1, 2, \ldots, M,\end{aligned} \tag{2}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm, $\mathbf{C} \in \mathbb{R}^{(N\times M)\times M}$ the coefficient matrix, $\mathbf{c}_i$ the i-th column of $\mathbf{C}$, and c_{ij} the element of $\mathbf{C}$ at the i-th row and j-th column.

As shown in [9] and [27], the representational flexibility of the patch-wise coding fashion makes LSC able to handle more complicated tracking scenarios than HSC. However, the holistic structural information is not exploited. As shown in Figure 2, the good and bad candidates have rather similarly small reconstruction errors since there are some atoms in the

dictionary that are very similar to the patches of both two candidates. Although Jia *et al*. [9] design an alignment pooling and Zhong *et al*. [22] propose a combination of HSC and LSC to handle this issue, the effectiveness is limited as the solutions are based on the representations that are obtained by LSC without the consideration of a spatial structure constraint on the used template patches. In contrast, we present the SALSC method which imposes both the global and local sparsity constraints when encoding a candidate. The SALSC method is therefore more suitable for visual tracking as it simultaneously takes the spatial structure information and representation flexibility into consideration.

III. Proposed Algorithm

In this section, we present the details of the proposed tracking method, which consists of three components: an appearance model, a template update strategy, and a decision model. We first introduce the SALSC model, which is designed to overcome the shortcomings of the existing LSC and HSC schemes for object tracking. Next, we describe our template update strategy, which is designed to maintain the effectiveness of the dictionary for modeling the target appearance over time. We then present the decision model which accurately selects the final tracked result from hundreds of target candidates.

A. Structure-Aware Local Sparse Coding

The central issue is how to carry out these two kinds of sparsities with the same dictionary in one coding procedure, as the global sparsity is generally implemented using templates as dictionary atoms, while the local sparsity is implemented using local patches of templates as dictionary atoms. To address this problem, we formulate our algorithm upon LSC, which has a natural ability of imposing the local sparsity constraints. To factor in the global sparsity constraint, we first design a rearranging function $f(\cdot)$, which rearranges every M rows of the coefficient matrix $\mathbf{C}$ into a new row as illustrated by Figure 3 so that each row corresponds to a different template. Then we apply the $\ell_{\infty,0}$ norm to the rearranged coefficient matrix $f(\mathbf{C})$ to count the number of nonzero rows, i.e. the number of used templates. The proposed model is formulated as

$$\arg\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}_l\mathbf{C}\|_F^2+\lambda_1\|f(\mathbf{C})\|_{\infty,0}+\lambda_2\sum_{i=1}^{M}\|\mathbf{c}_i\|_1$$
$$\text{s.t. } c_{ij}\geq 0 \quad \forall i=1,2,\ldots,N\times M,\ \forall j=1,2,\ldots,M, \quad (3)$$

In (3), the first and third terms measure the reconstruction error and local sparsity, respectively. In contrast to LSC, we introduce the second term to minimize the number of used templates, which implicitly enforces that the used dictionary atoms come from the templates that are more similar to the current considering candidate. Such a structure constraint is from a global view rather than a local spatial view and is used to address the shortcomings of traditional LSC models, which use patches from any templates to encode the target candidate and therefore reduce the discriminative ability of distinguishing the target from its backgrounds.

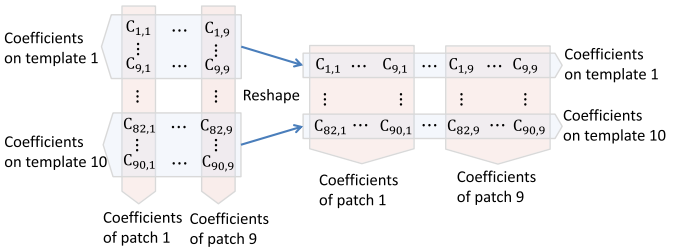


Fig. 3. The illustration of the rearranging function $f(\mathbf{C})$ with $N=10$ and $M=9$ (the same settings as in our experiments).

In the literature, there are several works which are also based on group sparsity such as [32]–[34]. Both [32] and [33] operate on full templates, but our model operates on local patches of templates. This leads to the differences in both the motivation and purposes of using the group sparsity among these methods. Specifically, [32] and [33] impose the group sparsity constraint in a supervised manner to learn a structured dictionary such that atoms in each group of the dictionary can represent images from the same class. By analyzing the indexes of non-zero coefficients, these models can be used for image classification. In contrast, our dictionary is composed of local patches that are directly sampled from target templates. Local patches from the same template are considered as a group. We use the group sparsity to implicitly enforce that the used local patches come from the templates that are more similar to the target candidate. The hierarchical sparse coding in [34] encodes one input signal each time, and the group sparsity is considered within coefficients of each input signal. Different from this, our model encodes M input signals each time, and the group sparsity is constrained across coefficients of all these local patches.

Solving SALSC: Finding the optimal solution for the objective function (3) is a NP-hard problem since the second term is non-convex [35]. In practice, the $\ell_{2,1}$ norm is often used to approximate the $\ell_{\infty,0}$ norm since their difference in effectiveness is subtle and sometimes the former behaves better as reported in [15]. Therefore, here we use the $\ell_{2,1}$ norm in this work:

$$\arg\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}_l\mathbf{C}\|_F^2+\lambda_1\|f(\mathbf{C})\|_{2,1}+\lambda_2\sum_{i=1}^{M}\|\mathbf{c}_i\|_1$$
$$\text{s.t. } c_{ij}\geq 0 \quad \forall i=1,2,\ldots,N\times M,\ \forall j=1,2,\ldots,M, \quad (4)$$

Due to the rearranging operation $f(\cdot)$ in (4), no existing solvers can be used for this optimization problem. We propose a fast solution based on the fast iterative shrinkage-thresholding algorithm (FISTA) [36] (although the APG [25] or ADMM [37] approach is also applicable). In general, the objective function $F(\mathbf{X})$ in FISTA is comprised of two parts,

$$\min\{F(\mathbf{X})=h(\mathbf{X})+g(\mathbf{X})\}, \quad (5)$$

where $h(\mathbf{X})$ is a smooth convex function with a Lipschitz continuous gradient over $\mathbf{X}$, and $g(\mathbf{X})$ is a continuous convex function over $\mathbf{X}$, which is possibly non-smooth. Given the values of $\mathbf{X}^k$, the weight variable t_k, and the intermediate

variable $\mathbf{Z}^k = \mathbf{X}^k$ at the k-th iteration, the general steps of FISTA are in the following iterative form

$$\mathbf{X}^{k+1} = \text{prox}_{t_k}(g)(\mathbf{Z}^k - \frac{1}{\Gamma}\nabla h(\mathbf{Z}^k)), \tag{6}$$

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \tag{7}$$

$$\mathbf{Z}^{k+1} = \mathbf{X}^{k+1} + (\frac{t_k - 1}{t_{k+1}})(\mathbf{X}^{k+1} - \mathbf{X}^k), \tag{8}$$

where Γ is the Lipschitz constant of gradient ∇h and the prox operation (6) is defined as

$$\text{prox}_t(g)(\mathbf{X}) = \arg\min_{\mathbf{U}} g(\mathbf{U}) + \frac{1}{2t}\|\mathbf{U} - \mathbf{X}\|_F^2. \tag{9}$$

To solve (4), we define

$$h(\cdot) = h(\mathbf{C}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{D}_l\mathbf{C}\|_F^2, \tag{10}$$

$$g(\cdot) = g(\mathbf{C}) = \lambda_1\|f(\mathbf{C})\|_{2,1} + \lambda_2\sum_{i=1}^{M}\|\mathbf{c}_i\|_1. \tag{11}$$

However, since the $\ell_{2,1}$ and ℓ_1 norms in (11) are two different matrices for $f(\mathbf{C})$ and $\mathbf{C}$, prox$(\cdot)$ of (6) cannot be directly solved. Fortunately, we observe that the second term in (11) computes the sum of the absolute values of all elements of $\mathbf{C}$, and this sum is not affected by the rearranging operation, which means

$$\lambda_2\sum_{i=1}^{M}\|\mathbf{c}_i\|_1 = \lambda_2\sum_{i=1}^{M}\|[f(\mathbf{C})]_i\|_1, \tag{12}$$

and hence we have

$$g(\mathbf{C}) = \lambda_1\|f(\mathbf{C})\|_{2,1} + \lambda_2\sum_{i=1}^{M}\|[f(\mathbf{C})]_i\|_1, \tag{13}$$

where the $\ell_{2,1}$ and ℓ_1 norms operate on the same matrix $f(\mathbf{C})$, which makes the prox (6) easy to solve with FISTA (e.g., using the SPAMS toolbox from http://spams-devel.gforge.inria.fr/). Based on (13), we have the new formulation of (6) as

$$\begin{aligned}\mathbf{Q} &= f(\mathbf{Z}^k - \frac{1}{\Gamma}\nabla h(\mathbf{Z}^k)),\\ \mathbf{Q}^* &= \text{prox}_{t_k}(g)(\mathbf{Q}),\\ \mathbf{C}^{k+1} &= f^{-1}(\mathbf{Q}^*),\end{aligned} \tag{14}$$

where $\nabla h(\mathbf{Z}^k) = \mathbf{D}_l^{\mathrm{T}}(\mathbf{D}_l\mathbf{Z}^k - \mathbf{Y})$ and $f^{-1}(\cdot)$ denotes the inverse operation of $f(\cdot)$. We summarize the main steps of the proposed optimization process in Algorithm 1.

Algorithm 1 Solving Structure-Aware Local Sparse Coding

Input: Candidate $\mathbf{Y}$, dictionary $\mathbf{D}_l$, maximum iterations K, initial solution $\mathbf{C}^0 = \mathbf{0}$, trade off parameters λ_1 and λ_2.
$\mathbf{Z}^1 = \mathbf{C}^0$;
for $k = 1$ *to* K **do**
  $\mathbf{Q} = f(\mathbf{Z}^k - \frac{1}{\Gamma}\nabla h(\mathbf{Z}^k))$;
  $\mathbf{Q}^* = \arg\min_{\mathbf{U}} \lambda_1\|\mathbf{U}\|_{2,1} + \lambda_2\sum_{i=1}^{M}\|\mathbf{U}_i\|_1 + \frac{1}{2t_k}\|\mathbf{U} - \mathbf{Q}\|_F^2 \quad \text{s.t.} \quad \mathbf{U}_{ij} \geq 0$;
  $\mathbf{C}^{k+1} = f^{-1}(\mathbf{Q}^*)$;
  $t^{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
  $\mathbf{Z}^{k+1} = \mathbf{C}^{k+1} + (\frac{t_k-1}{t^{k+1}})(\mathbf{C}^{k+1} - \mathbf{C}^k)$;
end
Output: $\mathbf{C}^{K+1}$.

B. Template Update

For sparse trackers, it is crucial to update target templates to adapt to appearance changes of the target over time.

Intuitively, the target appearance remains unchanged only for a certain period, and eventually the templates are no longer accurate for representing the target appearance. Therefore, the templates need to be updated over time to adapt to target appearance changes. On the other hand, if the templates are updated too frequently, small errors will be accumulated over time, which may eventually cause tracking drift.

To address this problem, we maintain a template set consisting of ten templates to reflect different target appearances. The template set is initialized as follows. The first template is the target region specified by the user; the remaining templates are created by perturbing a few pixels in four possible directions at the center location of the first template in the first frame. The template set is updated every five frames. Our update strategy keeps the first template unchanged throughout the video because it is the ground truth specified by the user and is helpful when the target appearance turns back from bad conditions such as occlusions and lighting changes. Unlike existing update strategies [9], [38] which update only one or all templates each time, we update three templates to avoid preserving too much old appearance information, which may not fully reflect the current appearances, or introducing too much new appearance information, which may not facilitate the sparse coding model when the target turns into old appearances. As shown in Figure 4, we update the templates at positions 2, 5, and 8, which cover both the old and new templates. We first delete templates at these three positions, then shift left the remaining seven templates, finally append new templates at the rear. Although updating templates at other positions is also possible, we find that updating templates at 2, 5, and 8 leads to a better tracking performance compared to that of updating templates at other positions such as (2, 3, 4), (4, 7, 10), or randomly three positions as shown in Figure 5.

We note the proposed update strategy can be used with most representation schemes, *e.g.*, SCM [22] and L1APG [21]. In this work, we use sparse representation and subspace learning to represent templates in a way similar to [9]:

$$\min_{\mathbf{z}}\|\mathbf{g} - \mathbf{B}\mathbf{z}\|_2^2 + \lambda_3\|\mathbf{z}\|_1, \tag{15}$$

where $\mathbf{B} = [\mathbf{E} \quad \mathbf{I}]$, $\mathbf{z} = [\mathbf{a}^\top \quad \mathbf{h}^\top]^\top$, and $\mathbf{g}$ is the observation vector. In addition, $\mathbf{E}$ denotes the matrix consisting of eigenbasis vectors of old templates. $\mathbf{I}$ and its corresponding coefficient vector $\mathbf{h}$ account for noise for the observation vector $\mathbf{g}$.

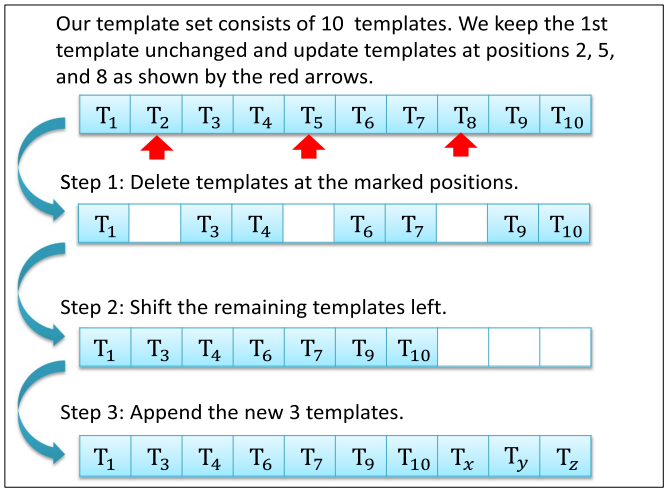


Fig. 4. Illustration of the proposed template update strategy, where $\mathbf{T}_x$, $\mathbf{T}_y$ and $\mathbf{T}_z$ denote new templates obtained using (15) and (16). Templates at the positions marked by red arrows are updated each time.

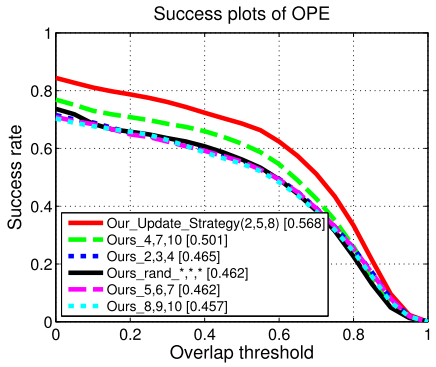


Fig. 5. Tracking results obtained by updating templates at different positions on the OTB50 dataset [26].

The new template is constructed by

$$\mathbf{T}_{\mathbf{new}} = \mathbf{E}\mathbf{a}. \tag{16}$$

C. Decision Model

As demonstrated in [39] and our experimental results in Figure 6, the reconstruction error alone is not effective for measuring similarity between candidate and target templates. We design a more reliable decision model which includes an SVM confidence score S_{svm}, a weighted pooling measurement S_{wp}, and a reconstruction error based measurement S_{re}. The confidence score of a candidate is computed by $\beta_1\ S_{svm} + \beta_2\ S_{wp} + \beta_3\ S_{re}$, where β_1, β_2, and β_3 are weight parameters.

1) SVM Confidence Score: Since we resize each candidate region to 32×32 pixels and sample patches of 16×16 pixels with a sliding window of 8 pixels, the four patches at the four corners encode the whole candidate. We concatenate coefficients of these four patches as the feature vector of a candidate, denoted as $\mathbf{b}$, *i.e.*, $\mathbf{b} = \phi(\mathbf{c}_1^\top, \mathbf{c}_3^\top, \mathbf{c}_7^\top, \mathbf{c}_9^\top)^\top$, where $\mathbf{c}_i$ denotes the i-th column of the coefficient matrix $\mathbf{C}$ and $\phi(\cdot)$ denotes the concatenation operation. The SVM confidence score of a candidate region is computed as

$$S_{svm} = \mathbf{w}^\top \mathbf{b}, \tag{17}$$

where $\mathbf{w}$ denotes the linear weights of an SVM classifier. To train the SVM classifier, at the starting frame we extract 10 positive samples around the initial annotation and 100 negative samples far away from the initial target position. To update the SVM model, we retrain it every 5 frames with new training data collected by sampling 10 positive examples and 100 negative examples in each of the latest 5 frames. The positive examples in the starting frame are also used to account for appearance changes.

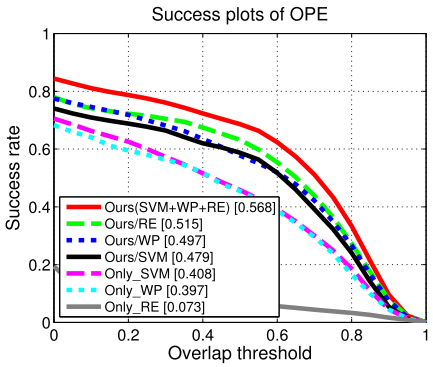


Fig. 6. Tracking results using different score measurements. The symbol '/' denotes 'without'.

2) Weighted Pooling Measurement: By solving the sparse coding model (4), we obtain the coefficient matrix $\mathbf{C}$, which denotes the sparse codes of local patches sampled from one candidate. $\mathbf{C}$ can be divided into N segments according to the template that each element of the matrix corresponds to, *e.g.* the first M row corresponds to the first template. These coefficient segments are summated together

$$\mathbf{S} = \sum_{i=1}^{N} \mathbf{C}^{(i)}, \tag{18}$$

where $\mathbf{C}^{(i)}$ denotes the submatrix of $\mathbf{C}$ from row $(i-1)\times M+1$ to row $i \times M$, which corresponds to the i-th template. Each element in $\mathbf{S}$ is the sum of coefficients of the template patches that are at the same position. With the sparsity assumption, each local patch within the target region can be best represented by the patches at the same positions of templates, and thus these templates patches have the largest coefficients. Therefore, the good candidate can be distinguished from the bad ones by comparing the sum of coefficients at the diagonal positions of $\mathbf{S}$

$$S_{wp} = \sum\nolimits_{i=1}^{M} \sum\nolimits_{j=1}^{M} [\mathbf{S} \circ \mathbf{W}]_{ij}, \tag{19}$$

where $\mathbf{W} \in \mathbb{R}^{M \times M}$ is a diagonal matrix with 1 on its diagonal positions. This operation is called *alignment-pooling* in [9]. In this paper, since the local patches are obtained in a sliding window fashion with half of the area overlapped, a candidate patch is encoded by template patches mainly sampled at the counterpart position and partly at its neighboring position. Thus, we propose a weighted pooling measurement to weight

positions differently. In particular, we define a weight matrix as

$$\mathbf{W}_{our} = \begin{bmatrix} 1 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.1 \\ 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

which puts weight 1 on coefficients at the counterpart position and 0.1 on the coefficient at a neighboring position. These weights are set empirically. The proposed weighted pooling measurement is obtained using (19) with such a weight matrix.

3) Reconstruction Errors Measurement: We define the reconstruction error based score S_{re} as

$$S_{re} = \sum_{i=1}^{M} 1/\|\mathbf{y}_i - \mathbf{D}\mathbf{c}_i\|_2^2. \tag{20}$$

D. Discussion

We discuss the differences between the proposed SALSC algorithm and the two most related sparse trackers, MTT [15] and SST [40]. The MTT method exploits structure information *among all target candidates* by using the $\ell_{2,1}$ norm to decide which candidates should be treated as a group and be represented by the same templates. As such, the MTT method needs to encode *all* candidates at the same time on the global level. The SST tracker extends the MTT method and uses local patches as encoding units. The SST approach emphasizes the *group property among all candidates* (similar to MTT) but on both the global and local levels. This is because the $\ell_{2,1}$ term in SST operates on patches of all candidates, of which the effect is *not* equivalent to the overall effect of separately exerting $\ell_{2,1}$ on patches of each candidate. In contrast, the SALSC algorithm does not consider the group property among candidates, and emphasizes similarity between *a candidate* and target templates on both the global and local levels. Namely, the SALSC algorithm aims to find its most similar templates from a template set and use patches of these templates to encode patches of this candidate. To this end, the SALSC approach encodes each candidate separately on the local patch level, and uses the $\ell_{2,1}$ norm on the rearranged coefficient matrix to constrain the number of used templates.

Overall, the MTT and SST methods focus on similarity between templates and *a group of candidates*, whereas the SALSC algorithm stresses similarity between templates and *a single candidate*. We present experimental evaluation on these methods in Section IV-F.

IV. Experiments

We first introduce the evaluation protocols and implementation details. Next, we examine the effectiveness of the proposed SALSC model and template update strategy. Finally, we report experimental evaluations of the proposed algorithm against the state-of-the-art tracking methods.

TABLE I
Challenge Attribute of Each UAV Sequence

Sequences	Challenges
bike3	ARC, CM, IV
boat2	ARC
boat3	ARC, BC, SOB
boat4	ARC, BC, POC, SOB
boat5	ARC, BC, SOB
car11	ARC, CM, FOC, IV, SOB, SV, VC
car12	ARC, BC, CM, FOC, SOB, VC
car13	ARC, CM, SV
person11	ARC, BC, IV, SOB, VC
person13	IV, OV, POC, SOB

A. Evaluated Methods

We compare the proposed algorithm with a total of 20 trackers: (i) recently published sparse coding based trackers: MASLA [41], LWIST [42], LRT [43], SST [40], SCM [22], ASLA [9], L1APG [44], and MTT [15]. (ii) recently published state-of-the-art trackers: Distractor-Aware Tracker (DAT) [45], Tracking with Gaussian Process Regression (TGPR) [2], Deep Learning Tracker (DLT) [4], Pyramid-Based Tracker(PBT) [46], Online Non-negative Dictionary Learning (ONNDL) [38], Superpixel Tracking (SPT) [47], Discriminative Scale Space Tracker (DSST) [48], Structured Output Tracking (Struck) [7], Context Tracker (CXT) [49], Tracking-Learning-Detection (TLD) [50], Distribution Fields for Tracking (DFT) [51], and Incremental Learning for Tracking (IVT) [14].

B. Evaluation Datasets

We evaluate the compared trackers on the tracking benchmark dataset OTB100 [27], which consists of 100 image sequences and involves 11 kinds of typical tracking challenging factors: illumination variation, scale changes, partial or full occlusion, non-rigid object shape deformation, motion blur, in-plane or out-of-plane rotation, out of view, background clutters, low resolution, and fast motion.

Moreover, we further test these trackers on 10 challenging Unmanned-Aerial-Vehicle (UAV) sequences [52] which cover challenges including Aspect Ratio Change (ARC), Camera Motion (CM), Illumination Variation (IV), Background Clutter (BC), Similar Object (SOB), Scale Variation (SV), Partial Occlusion (POC), Full Occlusion (FOC), Viewpoint Change, and Out-of-view (OV). A detailed challenge attribute of each sequence is summarized in Table I.

C. Evaluation Protocols

The performance of a tracker is measured by success plots in one-pass evaluation (OPE) and is ranked according to the area under curve (AUC) as specified in the tracking benchmark [26]. The success plots provide more information than the snap-short result by a single threshold of success rate. The success rate of a tracker is an average of overlap rates of all considered frames, and the overlap rate is computed by the intersection over the union of the tracked and the ground truth bounding boxes.

D. Implementation Details

We use grayscale images as the input for the proposed tracking algorithm. All target templates and candidate regions are normalized to 32×32 pixels from which a sliding window of size 16×16 with a sliding step of 8 pixels is used to sample patches in both the horizontal and vertical directions. The parameters α and K in Algorithm 1 are set to 0.015 and 10. The weights λ_1, λ_2, and λ_3 are all empirically set to 0.01. In addition, β_1, β_2, and β_3 in the decision model are set to 1, 0.1, and 0.01, respectively, according to their discriminative powers shown in Figure 6. All parameters are fixed for all experiments. The proposed tracker is implemented in MATLAB and it takes 0.48 seconds to process each frame on a 2.70 GHz Intel E5-2680 machine with 16G memory. The source code will be released to the public.

E. Ablation Analysis

We analyze the effectiveness of each component of the proposed tracking algorithm with different combinations. We use the LSC based ASLA [9] method as the baseline for evaluation. We denote the proposed SALSC model, update framework, and decision model as A, U, and D, respectively. The notation in the form of P $>$ Q is used to denote that the tracker P performs better than the tracker Q. The evaluated trackers are summarized below:

- Base+A: replacing the appearance model of the baseline method with A.
- Base+U: replacing the update strategy of the baseline method with U.
- Base+D: replacing the decision module of the baseline method with D.
- Base+AD: replacing the appearance and decision modules of the baseline method with A and D;
- Base+AU: replacing the appearance and update modules with A and U.
- Base+DU: replacing the decision and update modules with D and U.
- A+U+D: the combination used in the proposed tracking algorithm.

In addition, we evaluate the proposed template update method with two commonly used methods proposed in [9] and [38]. Comparisons of different template update strategies are presented in Figure 7 and the AUC rankings (a bigger score is better) of overall performance are shown in Figure 8.

All the three components of the proposed tracker contribute to more accurate tracking results. First, Base+A $>$ Base and Base+AD $>$ Base+D in all tracking sequences indicate that the proposed SALSC method extracts more discriminative features than the LSC approach regardless decision modules. Second, in the presence of most challenging factors, Base+U $>$ Base and Base+AU $>$ Base+A show that the proposed template update strategy contributes to the higher tracking performance. Third, Base+D $<$ Base, Base+DU $<$ Base but Base+AD $>$ Base+A indicates that our decision model performs badly when collaborating with the traditional local sparse coding model which is used by Base but performs well

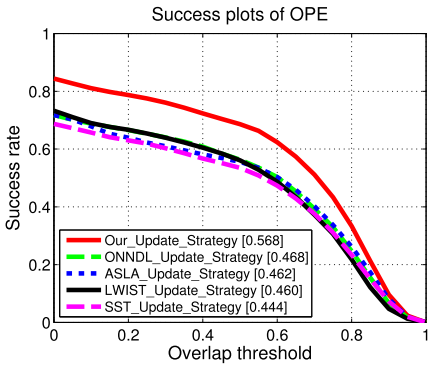


Fig. 7. Tracking results using different template update strategies.

in achieving better results when cooperating with the proposed SALSC appearance model.

We further evaluate the proposed template update strategy with comparison to other four commonly used update schemes in Figure 7. The ASLA_Update_Strategy in [9] randomly updates one template each time. The ONNDL_Update_Strategy in [38] updates all templates each time. The SST_Update_Strategy in [40], [44], and [56] assigns a weight to each template to indicate how representative the template is. The template with the lowest weight is replaced by the current tracking result. The LWIST_Update_Strategy in [42] incrementally updates the templates whose ℓ_2 distances to the current tracked result are smaller than a given threshold. As shown in Figure 7, the proposed template update strategy achieves about 10% improvement compared to other four update schemes, which demonstrates the effectiveness of our template update strategy.

Among the three components of the proposed tracker, the SALSC module contributes the most, then the update strategy, and the decision model. In terms of occlusion, out of view, fast motion, and deformation, we observe that Base+A $>$ Base +U and Base+AD $>$ Base+DU, which show that SALSC scheme contributes more than the update module. However, in terms of motion blur, illumination variation, rotation, background clutters, low resolution, and scale variation, the results show that Base+U $>$ Base+A and Base+DU $<$ Base+AD. Namely, the decision module also plays an important role. Furthermore, the result Base+DU $<$ Base+AD indicates that the SALSC plays a larger role than the update module. In most sequences, we observe Base+U $>$ Base+D and Base+AU $>$ Base+AD, which show that our update strategy contributes more than the decision model, regardless of whether the SALSC or LSC scheme is used to model object appearance.

The results show that Base+A performs better than Base in all sequences. In particular, in the face of occlusion, out-of-view, fast motion, and deformation, Base+A achieves 10 percent improvement over the baseline method. These results demonstrate the effectiveness of the proposed scheme that exploits both global and local sparse coding.

F. Evaluation Against the State-of-the-Art Methods

1) Quantitative Evaluation: Figure 9 shows the overall performance of all the evaluated trackers on the

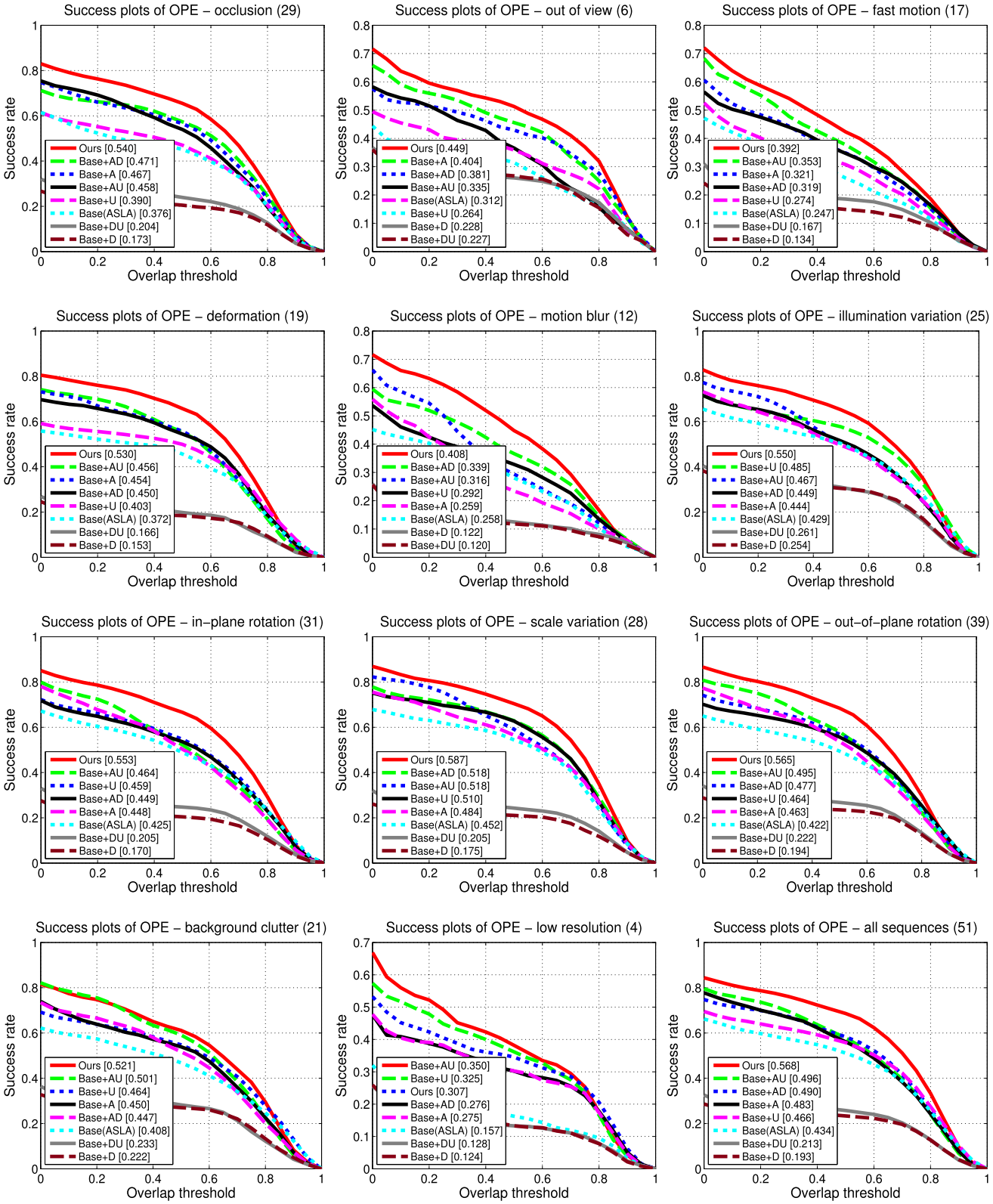


Fig. 8. Tracking results based on 11 attribute challenges where A, U, and D denote the proposed SALSC based appearance model, template update strategy, and decision model, respectively. The last subfigure shows the overall performance of evaluated tracking methods.

OTB100 dataset [27]. For presentation clarity, only the top 10 performing trackers are shown. The trackers in Precision plots are ranked according to the success rate with the center location error at a threshold of 20 pixels. The trackers in Success plots are ranked according to the area under the curve (AUC) with the overlap threshold from 0 to 1. For

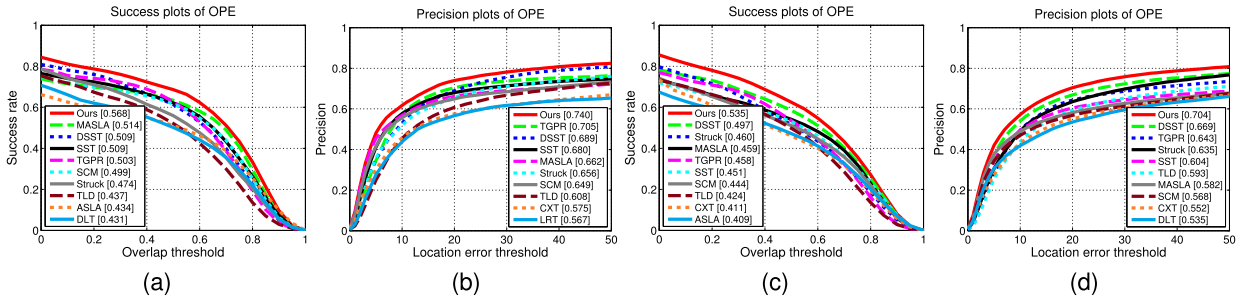


Fig. 9. (a) and (b) show the tracking performance of 21 trackers on the OTB50 dataset [26]. (c) and (d) show the tracking performance on the OTB100 dataset [27]. Only the top 10 performing trackers are presented for clarity.

TABLE II

AUC RANKINGS ON 100 IMAGE SEQUENCES CALCULATED IN DIFFERENT OVERLAP RATE RANGES

0.9∼1.0	0.8∼1.0	0.7∼1.0	0.6∼1.0	0.5∼1.0	0.4∼1.0	0.3∼1.0	0.2∼1.0	0.1∼1.0	0∼1.0
Ours[0.033]	**Ours[0.109]**	**Ours[0.191]**	**Ours[0.264]**	**Ours[0.327]**	**Ours[0.380]**	**Ours[0.426]**	**Ours[0.467]**	**Ours[0.503]**	**Ours[0.535]**
SST[0.032]	DSST[0.100]	DSST[0.177]	DSST[0.245]	DSST[0.304]	DSST[0.354]	DSST[0.397]	DSST[0.435]	DSST[0.468]	DSST[0.497]
ASLA[0.031]	SST[0.096]	MASLA[0.165]	MASLA[0.231]	MASLA[0.285]	MASLA[0.329]	MASLA[0.368]	MASLA[0.401]	MASLA[0.431]	Struck[0.460]
DSST[0.029]	MASLA[0.088]	SST[0.163]	SST[0.224]	SST[0.277]	SST[0.321]	SST[0.359]	SST[0.393]	TGPR[0.426]	MASLA[0.459]
SCM[0.027]	SCM[0.087]	SCM[0.156]	SCM[0.216]	SCM[0.267]	SCM[0.310]	TGPR[0.349]	TGPR[0.390]	Struck[0.425]	TGPR[0.458]
MASLA[0.024]	ASLA[0.085]	ASLA[0.146]	ASLA[0.200]	Struck[0.251]	TGPR[0.300]	SCM[0.348]	Struck[0.387]	SST[0.423]	SST[0.451]
CXT[0.024]	CXT[0.077]	CXT[0.138]	Struck[0.196]	ASLA[0.247]	Struck[0.300]	Struck[0.345]	SCM[0.382]	SCM[0.414]	SCM[0.444]
L1APG[0.024]	L1APG[0.075]	Struck[0.135]	CXT[0.192]	TGPR[0.246]	ASLA[0.288]	ASLA[0.323]	TLD[0.358]	TLD[0.393]	TLD[0.424]
DLT[0.022]	Struck[0.073]	DLT[0.128]	TGPR[0.187]	CXT[0.238]	CXT[0.279]	TLD[0.318]	ASLA[0.354]	ASLA[0.382]	CXT[0.411]
Struck[0.022]	DLT[0.072]	L1APG[0.127]	DLT[0.179]	DLT[0.224]	TLD[0.274]	CXT[0.315]	CXT[0.348]	CXT[0.380]	ASLA[0.409]

TABLE III

AVERAGE SUCCESS RATE. THE BEST TWO RESULTS ARE SHOWN IN RED BOLD AND BLUE ITALIC FONTS, RESPECTIVELY

	Ours	ASLA	SCM	LRT	DSST	TGPR	SST	LWIST	MASLA
bike3	*0.162*	0.160	0.147	0.127	**0.165**	0.130	0.064	0.155	0.160
boat2	0.895	0.853	0.878	0.921	0.768	0.630	*0.940*	**0.941**	0.832
boat3	**0.658**	0.621	0.524	*0.633*	0.513	0.494	0.549	0.269	0.590
boat4	*0.697*	0.641	0.586	0.597	0.654	0.530	**0.759**	0.542	0.594
boat5	*0.573*	0.558	0.453	0.490	0.571	0.431	**0.673**	0.251	0.497
car11	**0.068**	*0.064*	0.045	0.060	0.041	0.034	0.040	0.049	0.050
car12	*0.047*	**0.051**	0.023	0.021	0.022	0.020	0.020	0.024	0.022
car13	*0.630*	0.543	0.238	0.433	0.009	0.004	0.173	0.580	**0.637**
person11	0.614	0.538	0.439	0.621	0.497	*0.625*	**0.631**	0.197	0.614
person13	**0.621**	0.320	0.280	0.020	0.404	0.019	0.019	0.019	*0.568*

completeness, we also provide more detailed AUC rankings with all-round overlap rate ranges in Table II.

Overall, the proposed tracking algorithm performs favorably against the state-of-the-art methods. Compared to sparse trackers, such as MASLA, SCM, SST, ASLA, LWIST, the proposed algorithm achieves about 7%, 9%, 8%, 13%, and more than 13% improvement, respectively, as shown in the last column of Table II. In addition, we provide more detailed results in terms of the average center location error and average overlap rate [54] on 10 representative sequences in Table IV and Table V. The results show that the proposed tracking algorithm obtains higher tracking success rates and smaller distance errors on most of these sequences than the compared state-of-the-art trackers.

Table III presents the tracking results in term of the average success rate on each of the 10 challenging UAV sequences. The proposed algorithm ranks top two on most of these sequences compared to the state-of-the-art sparse trackers such as MASLA, SST, LRT, LWIST, and LRT.

2) Qualitative Evaluation: Figure 10 shows some tracking results on challenging sequences. We discuss the performance of the evaluated tracking methods in presence of different challenging factors.

a) Occlusion and deformation: In the *basketball* sequence, the target person undergoes heavy occlusion at frame 16 and large deformation throughout the sequence. The Struck method loses the target from frame 30. The ASLA, SCM, and DSST algorithms drift away at frame 593 and frame 706, while the proposed tracker locates the target accurately. This can be attributed to the SALSC scheme can exclude dissimilar candidate regions on the global level and correctly encode the unoccluded and undeformed parts on the local level.

b) Occlusion and background clutter: In the *football* sequence, at frame 286 the target player is heavily occluded. Only the DSST and proposed trackers locate the target accurately as shown in Figure 10. In this case, the proposed SALSC scheme contributes the most as demonstrated in Figure 8 as it can exclude dissimilar candidate regions on the global level and capture the unoccluded regions on the local level. The proposed update strategy also plays an important role in such scenarios as illustrated in Figure 8 (occlusion and background clutter).

c) Illumination and rotation: Throughout the *skating* sequence, the target skater undergoes in-plane and out-of-plane rotations and the illumination changes drastically at frame 304. All trackers except the proposed method lose the target after frame 304. Similarly in the *car4* sequence, the proposed template update strategy plays the most important role, as demonstrated in Figure 8 (illumination and rotation).

Fig. 10. Tracking results of the proposed algorithm against six state-of-the-art trackers on challenging image sequences.

TABLE IV

AVERAGE SUCCESS RATE. THE BEST TWO RESULTS ARE SHOWN IN RED BOLD AND BLUE ITALIC FONTS, RESPECTIVELY

	TLD	LWIST	MASLA	IVT	DLT	CXT	DAT	SPT	ASLA	MTT	L1APG	SCM	Struck	TGPR	DSST	Ours
car4	0.63	0.87	0.85	*0.88*	0.86	0.31	0.01	0.23	0.75	0.45	0.25	0.76	0.49	0.50	0.69	**0.89**
carDark	0.45	0.78	0.79	0.66	0.66	0.57	0.03	0.09	0.85	0.83	0.88	0.84	**0.89**	0.81	0.59	*0.86*
david2	0.69	0.52	0.86	0.70	0.81	*0.88*	0.36	0.53	**0.90**	0.86	0.86	0.75	0.87	0.82	0.48	**0.90**
doll	0.57	0.28	*0.83*	0.43	*0.83*	0.75	0.39	0.49	*0.83*	0.39	0.45	*0.83*	0.55	0.58	*0.83*	**0.86**
freeman3	0.45	0.50	0.60	0.39	0.73	0.71	0.30	0.39	*0.75*	0.46	0.35	0.72	0.26	0.05	0.35	**0.80**
freeman4	0.22	0.26	0.11	0.15	0.33	0.17	0.23	0.02	0.13	0.22	*0.34*	0.27	0.17	0.27	0.13	**0.41**
girl	0.57	0.35	0.69	0.17	0.54	0.55	0.46	0.14	0.71	0.67	0.73	0.68	*0.75*	0.67	0.40	**0.77**
shaking	0.39	0.04	0.04	0.03	0.06	0.12	0.05	0.10	0.46	0.04	0.08	*0.69*	0.35	0.49	0.66	**0.77**
singer2	0.22	0.47	0.76	0.04	0.04	0.07	0.02	0.63	0.04	0.04	0.04	0.17	0.04	*0.73*	0.04	**0.75**
walking2	0.31	0.67	*0.83*	0.80	0.81	0.37	0.24	0.32	0.37	0.79	0.76	0.82	0.51	0.60	0.80	**0.84**

TABLE V

AVERAGE CENTER LOCATION ERROR (IN PIXELS). THE BEST TWO RESULTS ARE SHOWN IN RED BOLD AND BLUE ITALIC FONTS, RESPECTIVELY

	TLD	LWIST	MASLA	IVT	DLT	CXT	DAT	SPT	ASLA	MTT	L1APG	SCM	Struck	TGPR	DSST	Ours
car4	12.8	2.4	2.9	2.1	2.7	58.1	155.5	82.1	**1.6**	22.3	77.0	4.3	8.7	6.1	8.3	*1.9*
carDark	27.5	2.1	1.8	8.4	1.4	16.5	104.7	142.4	1.5	1.6	**1.0**	*1.3*	rbf1.0	2.1	5.8	*1.3*
david2	5.0	3.2	*1.0*	1.2	1.9	1.3	14.2	5.5	1.5	1.7	1.4	3.4	1.5	2.1	7.9	**0.9**
doll	6.0	48.2	**2.9**	32.7	5.5	4.7	26.8	40.2	11.8	110.3	5.9	3.4	8.9	6.0	**2.9**	*3.0*
freeman3	29.3	14.3	5.3	35.8	*3.1*	3.6	43.2	73.9	3.2	15.6	33.1	3.2	16.8	88.3	15.8	**2.0**
freeman4	39.2	39.2	101.2	43.0	35.1	65.6	32.0	56.5	70.2	23.5	*22.1*	37.7	48.7	48.1	68.2	**14.2**
girl	9.8	10.6	3.1	22.5	9.9	11.0	15.2	78.5	3.3	4.3	2.8	*2.6*	*2.6*	7.7	11.0	**2.3**
shaking	37.1	108.4	101.9	85.7	114.4	129.2	116.2	115.5	22.4	97.9	109.8	*11.0*	30.7	18.9	12.6	**7.4**
singer2	58.3	34.4	*10.4*	175.5	181.9	163.6	246.5	16.2	175.3	209.7	180.9	113.6	174.3	**10.1**	181.1	10.6
walking2	44.6	2.9	2.1	2.5	2.2	34.7	23.1	126.2	37.4	4.0	5.1	*1.7*	11.2	5.9	2.4	**1.5**

The proposed update strategy can incorporate the latest observations with the existing templates effectively to account for large appearance changes.

d) Scale: The *singer1* and *car4* sequences contain significant scale variations, and the TGPR, SCM, and Struck methods do not adapt to the target scale well. As shown in Figure 8 (scale), our tracker retains templates effectively to account for appearance changes. The DAT tracker loses track of the target object as the color histogram features are not effective in these scenarios.

e) Scale, rotation, background clutter, and illumination variation: The *shaking* sequence is challenging as the target

object undergoes large scale and rotation changes in a cluttered background with varying illumination. From frame 165, all methods except the DSST and proposed trackers drift away from the target as shown in Figure 10(e). Table IV and Table V show that the proposed tracker performs favorably against the other methods.

V. Conclusions

In this work, we propose a structure-aware local sparse coding model for effective object tracking. By considering the global and local sparsity constraints, the proposed representation is more discriminative than existing approaches. In addition, we propose a fast optimization method to compute sparse coefficients, and an effective template update strategy to account for appearance changes. Extensive experimental results with comparisons to the state-of-the-art methods on a large tracking benchmark dataset demonstrate the effectiveness of the proposed tracking algorithm.

References

[1] 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.

[2] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[3] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 47–56.

[4] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[5] G. Li, L. Qin, Q. Huang, J. Pang, and S. Jiang, "Treat samples differently: Object tracking with semi-supervised online covboost," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 627–634.

[6] Y. Lu, T. Wu, and S.-C. Zhu, "Online object tracking, learning and parsing with and-or graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3462–3469.

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

[8] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 142–149.

[9] 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.

[10] J. Kwon and K. M. Lee, "Interval tracker: Tracking by interval analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3494–3501.

[11] S. Zhang, X. Lan, Y. Qi, and P. C. Yuen, "Robust visual tracking via basis matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 421–430, Mar. 2017.

[12] D.-H. Kim, H.-K. Kim, S.-J. Lee, W.-J. Park, and S.-J. Ko, "Kernel-based structural binary pattern tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 8, pp. 1288–1300, Aug. 2014.

[13] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. CVPR*, 2016, pp. 4303–4311.

[14] 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.

[15] 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.

[16] A. Li and S. Yan, "Object tracking with only background cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1911–1919, Nov. 2014.

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

[18] 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.

[19] 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.*, Jun. 2015, pp. 597–606.

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

[21] X. Mei and H. Ling, "Robust visual tracking using l_1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1436–1443.

[22] 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.

[23] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. A. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 624–637.

[24] B. Liu, J. Huang, C. Kulikowski, and L. Yang, "Robust visual tracking using local sparse appearance model and K-selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2968–2981, Dec. 2013.

[25] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Optim.*, vol. 6, pp. 615–640, 2010.

[26] 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.

[27] 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.

[28] D. Yadin, "The neurobiology of consolidations, or, how stable is the engram?" *Annu. Rev. Psychol.*, vol. 55, pp. 51–86, Feb. 2004.

[29] N. Danko and S. Wolf, "Creation of visual long-term memory," *Perception Psychophys.*, vol. 69, no. 6, pp. 904–912, 2007.

[30] N. Axmacher, S. Haupt, M. X. Cohen, C. E. Elger, and J. Fell, "Interference of working memory load with long-term memory formation," *Eur. J. Neurosci.*, vol. 29, no. 7, pp. 1501–1513, 2009.

[31] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 46, no. 7, pp. 1772–1788, Jul. 2013.

[32] Y. Zhang, Z. Jiang, and L. S. Davis, "Learning structured low-rank representations for image classification," in *Proc. CVPR*, 2013, pp. 676–683.

[33] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. CVPR*, 2011, pp. 1697–1704.

[34] R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach, "Proximal methods for hierarchical sparse coding," *J. Mach. Learn. Res.*, vol. 12, pp. 2297–2334, Jul. 2011.

[35] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, 2006.

[36] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[38] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 657–664.

[39] 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.

[40] T. Zhang *et al.*, "Structural sparse tracking," in *Proc. CVPR*, Jun. 2015, pp. 150–158.

[41] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via coarse and fine structural local sparse appearance models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4555–4564, Oct. 2016.

[42] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, Sep. 2015.

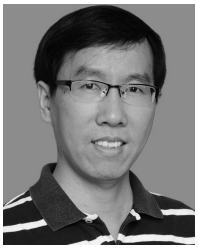
[43] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem, "Robust visual tracking via consistent low-rank sparse learning," *Int. J. Comput. Vis.*, vol. 111, no. 2, pp. 171–190, 2015.

[44] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1830–1837.
[45] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2113–2120.
[46] Z. Zhang and K. H. Wong, "Pyramid-based visual tracking using sparsity represented mean transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1226–1233.
[47] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.
[48] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.
[49] T. B. Dinh, N. Vo, and G. G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1177–1184.
[50] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. CVPR*, Jun. 2010, pp. 49–56.
[51] L. Sevilla-Lara and E. G. Learned-Miller, "Distribution fields for tracking," in *Proc. CVPR*, 2012, pp. 1910–1917.
[52] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. ECCV*, 2016, pp. 445–461.
[53] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.
[54] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

Yuankai Qi received the B.S. and M.S. degrees from the Harbin Institute of Technology, China, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include object tracking, video segmentation, sparse coding, and machine learning. He serves as a Reviewer for top journals, such as the IEEE TIP, TMM, and TCSVT.

Lei Qin received the Ph.D. degree in computer science from the Institute of Computing Technology of the Chinese Academy of Sciences (ICTCAS), in 2008. He is currently an Associate Professor with the Key Laboratory of Intelligent Information Processing, ICTCAS. He has authored over 50 academic papers. He is a Reviewer for the IEEE TMM, TCSVT, and TCYB. His research interests include image/video processing, computer vision, and pattern recognition. He has served as a Technical Program Committee Member for various international conferences, including ECCV, ICPR, ICME, PSIVT, ICIMCS, and PCM.

Jian Zhang received the B.Sc. degree in mathematics and the M.Eng. and Ph.D. degrees in computer science and technology from the Harbin Institute of Technology, in 2007, 2009, and 2014, respectively. He is currently a Post-Doctoral Fellow with the Visual Computing Center, King Abdullah University of Science and Technology, Saudi Arabia. His research interests include image/video restoration and compression, optimization, and deep learning. He was a recipient of the Best Paper Award and Best Student Paper Award at the IEEE International Conference on Visual Communication and Image Processing in 2011 and 2015, respectively.



Shengping Zhang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He was a Post-Doctoral Research Associate with Brown University and with Hong Kong Baptist University, and a Visiting Student Researcher with the University of California at Berkeley. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology in Weihai. He has authored or co-authored over 50 research publications in refereed journals and conferences. His research interests include deep learning and its applications in computer vision. He is also an Associate Editor of *Signal Image and Video Processing*.

Qingming Huang (F'18) received the B.S. and Ph.D. degrees from the Harbin Institute of Technology (HIT), in 1988 and 1994, respectively. He is currently a Professor of computer science and technology with HIT and the Deputy Dean of computer and control engineering with the University of Chinese Academy of Sciences. He has authored over 300 academic papers in international journals, such as the IEEE TIP, TKDE, TMM, and TCSVT, and top-level international conferences, including the ACM Multimedia, ICCV, CVPR, ECCV, VLDB, AAAI, and IJCAI. His research interests include multimedia computing, image/video processing, pattern recognition, and computer vision.

Ming-Hsuan Yang (SM'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2000. He is currently a Professor of electrical engineering and computer science with the University of California at Merced, Merced, CA, USA. He received the NSF CAREER Award in 2012 and the Google Faculty Award in 2009. He is a Senior Member of the ACM. He served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, the *International Journal of Computer Vision*, *Image and Vision Computing*, and the *Journal of Artificial Intelligence Research.*