Visual Tracking Via Sparse Representation With Reliable Structure Constraint

Jie Guo, Tingfa Xu, Ziyi Shen, and Guokai Shi

***Abstract*—In this letter, we present a novel visual tracking algorithm based on sparse representation. In contrast to just use the target templates and the trivial templates to sparsely represent the target, we propose to further constrain the model with a set of discriminative weight maps. These weight maps contain the reliable structures of the target object. They help the model penalize the trivial template coefficients depending on the reliable structures of the target object. Then, the target object can be well represented by a sparse set of target templates together with a sparse set of target weight maps. We propose a unified objective function to integrate these two sparse representation problems together. This optimization problem can be well solved by the proposed iteration manner and a customized accelerated proximal gradient method. Furthermore, a novel weight map constructing method is proposed based on consistent motion property and forward–backward errors. Plenty of qualitative and quantitative evaluations demonstrate that our method performs favorably against the state-of-the-art methods in a wide range of tracking scenarios.**

***Index Terms*—Reliable structures, sparse representation, visual tracking, weight maps.**

I. Introduction

ONLINE visual tracking remains a very difficult problem because of the complicated tracking scenarios (e.g., rotations, scale changes, occlusions, deformations, background clutters).

To solve this tough problem, recently, many remarkable sparse representation based trackers have been proposed [1]. Mei and Ling [2] propose a novel l_1 tracker, which uses a set of target image patches as target templates and a set of identity pixel basis as trivial templates. By the sparsity constraints, the target patch can be sparsely represented by all these templates. However, the identity pixel basis can be activated to represent any image patches, resulting in failures in discriminating the target from background. To solve this problem, Bao *et al*. [3] propose to use an L2 norm regularization term on the coefficients of the trivial templates when there is no occlusion. Furthermore, Zhang *et al*. [4] propose a multitask tracking algorithm, which explores the joint-sparsity property of all candidates. All these trackers have improved the original l_1 tracker from different aspects. They treat all pixels equally in the model. This loses the spatial information of the target and ignores the differences among different pixels. The trivial templates are activated to represent the noises of target patch, which means we should use a weight map to penalize more on pixels which belong to target object and penalize less on pixels which belong to background and occluded parts. Zhang *et al*. [5] use a weight matrix in their model to balance the importance among different pixels within a weighted least squares framework. Other methods [6]–[13] use different kinds of image patches or the learned variation dictionary to replace the identity pixel basis. These methods, on the other side, cannot make full use of each pixel.

This letter presents an efficient tracking approach based on sparse representation. The main contributions of this work are as follows. First, we propose to use a set of weight maps to further regularize the coefficients of the trivial templates. The weight maps preserve the reliable structures of the target object and the background. This makes the model more robust and discriminative by the constraint of the reliable structures. Second, we propose a unified objective function to integrate the problems of solving the coefficients of target templates and weight maps together. This optimization problem can be well solved by the proposed iteration manner and a customized accelerated proximal gradient (APG) method. Third, we propose a novel method based on the pyramidal Lucas–Kanade tracker [14] to construct the weight maps.

II. Object Tracking Based on Sparse Representation With Reliable Structure Constraint

A. Sparse Representation With Reliable Structure Constraint

Motivated by [2], [3], and [15], we assume that the target observation $\mathbf{y}$ can be sparsely represented by the target template set $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{\boldsymbol{m}}] \in R^{N\times m}$ and the trivial template set $\mathbf{I} \in R^{N\times N}$, where N is the dimension of the observation vector, m is the number of target templates and $\mathbf{I}$ is an identity matrix. Unlike all the other sparse representation based methods, our method exploits the reliable structures of object target. Then, we construct a set of weight map templates $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{w}_{n+1}, \ldots \mathbf{w}_{2n}] \in R^{N\times 2n}$ based on these structures, where n (in this letter we set it to be 40) is the number of target weight maps. The first half of $\mathbf{W}$ is target weight maps, the second half is the inverse patterns of the target weight maps as illustrated in Fig. 1. Details about the constructing of weight maps are described in Section II-B. The model can separately control the regularization amount of different pixels with

Manuscript received October 2, 2016; revised November 12, 2016; accepted December 21, 2016. Date of publication December 28, 2016; date of current version January 12, 2017. This work was supported in part by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802, and in part by the General Program of National Nature Science Foundation of China under Grant 61371132 and Grant 61471043. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xudong Jiang.

The authors are with the School of Optical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: jieguo_2013@163.com; tfxu_bit@163.com; joanshen0508@gmail.com; 451942036@qq.com).

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

Digital Object Identifier 10.1109/LSP.2016.2645819

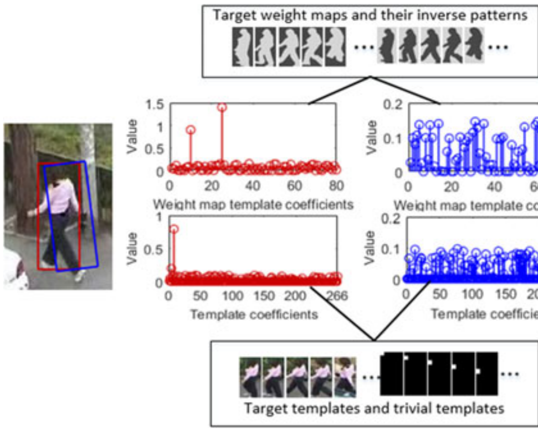

Fig. 1. Good target candidate can be well approximated by a sparse set of target templates together with a sparse set of target weight maps, whereas the bad target candidate cannot be well represented anyway.

a weight map. Then, we get the following objective function:

$$\underset{\mathbf{a}}{\operatorname{argmin}}\ \|\mathbf{y}-\mathbf{B}\mathbf{a}\|_2^2+\lambda\|\mathbf{a}_T\|_1+\lambda\|\mathbf{w}\odot\mathbf{a}_I\|_1,\quad \text{s.t. } \mathbf{a}_T\geq 0 \tag{1}$$

where $\mathbf{B}=[\mathbf{T},\mathbf{I}]$, $\mathbf{a}=[\mathbf{a}_T;\mathbf{a}_I]$, and $\mathbf{w}\in\mathbf{W}$ is a vectorized weight map, λ controls the amount of regularization. Operator $\odot$ denotes element-wise multiplication. Let $\tilde{\mathbf{a}}_I=\mathbf{w}\odot\mathbf{a}_I$, then (1) can be expressed as

$$\underset{\mathbf{a}}{\operatorname{argmin}}\ \left\|\mathbf{y}-\mathbf{T}\mathbf{a}_T-\mathbf{I}\frac{\tilde{\mathbf{a}}_I}{\mathbf{w}}\right\|_2^2+\lambda\|\mathbf{a}\|_1,\quad \text{s.t. } \mathbf{a}_T\geq 0 \tag{2}$$

where $\mathbf{a}=[\mathbf{a}_T;\tilde{\mathbf{a}}_I]$ and $-$ denotes element-wise division in this equation.

In (1) and (2), the weight map $\mathbf{w}$ is supposed to accurately reflect the reliable structure of the target in current frame, but it is difficult to obtain such a weight map. However, the reliable structures of target object have the same property with the target itself, i.e., the reliable structure of current target can be sparsely represented by the reliable structure templates (i.e., the weight map templates). Thus, we remain a set of weight map templates $\mathbf{W}$. Then, we present a joint sparse objective function

$$\begin{aligned}J\{\mathbf{a},\mathbf{m}\}\ &=\|\mathbf{y}-\mathbf{T}\mathbf{a}_T-\mathcal{D}(\mathbf{W}_r\mathbf{b})\,\tilde{\mathbf{a}}_I\|_2^2+\lambda\|\mathbf{a}\|_1+\mu\|\mathbf{b}\|_1\\ &\text{s.t. } \mathbf{a}_T\geq 0 \text{ and } \mathbf{b}\geq 0\end{aligned} \tag{3}$$

where $\mathbf{a}=[\mathbf{a}_T;\tilde{\mathbf{a}}_I],\mathbf{b}=[b_1,\ldots,b_{2n}]^{\mathrm{T}}\in R^{2n}$ is the sparse coefficients of the weight maps, parameter μ controls the amount of regularization of $\mathbf{b}$, and $\mathcal{D}(\mathbf{x})$ represents the diagonal matrix of which the diagonal elements are formed by $\mathbf{x}$. Matrix $\mathbf{W}_r=[\mathbf{w}_r^1,\ldots,\mathbf{w}_r^{2n}]=[\frac{1}{\mathbf{w}_1},\ldots,\frac{1}{\mathbf{w}_{2n}}]$, where $1\in R^N$ is a column vector whose entries are all ones and $-$ denotes element-wise division in this equation. The estimated coefficients $\mathbf{a}$ and $\mathbf{b}$ can be achieved by minimizing the objective function (3) with non-negativity constraints:

$$\left\{\hat{\mathbf{a}},\hat{\mathbf{b}}\right\}=\underset{\mathbf{a},\mathbf{b}}{\operatorname{argmin}}\ J\{\mathbf{a},\mathbf{b}\}\quad \text{s.t. } \mathbf{a}_T\geq 0 \text{ and } \mathbf{b}\geq 0. \tag{4}$$

In this optimization problem, coefficients $\mathbf{a}$ and $\mathbf{b}$ are all unknown, making the solution of this problem intractable. In this work, we present an iteration method to search the minima of the optimization problem (4). The initiation of coefficient $\mathbf{a}$ is obtained by solving (1) with all the elements of $\mathbf{w}$ set to be 1.

Algorithm 1: Fast numerical algorithm for solving problem (6).

Set $a_0=a_{-1}=0\in R^N$ and set $\rho_0=\rho_{-1}=1$.
For $k=0,1,\ldots,$ until converge or a maximal number of iterations have been met
$\beta_{k+1}=a_k+\frac{\rho_{k-1}-1}{\rho_k}(a_k-a_{k-1})$
$g_{k+1}|_T=\beta_{k+1}|_T-(\mathbf{P}^{\mathrm{T}}(\mathbf{P}\beta_{k+1}-\mathbf{y}))|_T/L-\lambda 1_T/L$
$g_{k+1}|_I=\beta_{k+1}|_I-(\mathbf{P}^{\mathrm{T}}(\mathbf{P}\beta_{k+1}-\mathbf{y}))|_I/L$
$a_{k+1}|_T=\max(0,\ g_{k+1}|_T)$
$a_{k+1}|_I=S_{\lambda/L}(g_{k+1}|_I)$
$\rho_{k+1}=(1+\sqrt{1+4\rho_k^2})/2$
End
Obtain $\hat{\mathbf{a}}$ via $\hat{\mathbf{a}}=a_{k+1}$.

Then coefficients $\mathbf{a}$ and $\mathbf{b}$ can be achieved by iteratively solving subproblems (a) and (b):

a) Fix $\mathbf{a}$, solve $\mathbf{b}$: if $\mathbf{a}$ is given, problem (4) turns into a standard l_1-regularization problem, i.e.,

$$\begin{aligned}&\underset{\mathbf{b}}{\operatorname{argmin}}\ \|\mathbf{y}-\mathbf{T}\mathbf{a}_T-(\mathbf{W}_r\odot\mathcal{C}(\tilde{\mathbf{a}}_I))\,\mathbf{b}\|_2^2+\mu\|\mathbf{b}\|_1\\ &\qquad\text{s.t. } \mathbf{b}\geq 0\end{aligned} \tag{5}$$

where $\mathcal{C}(\tilde{\mathbf{a}}_I)=[\tilde{\mathbf{a}}_I,\ldots,\tilde{\mathbf{a}}_I]\in R^{N\times N}$. This problem can be solved by the least absolute shrinkage and selection operator (LASSO) method [16].

b) Fix $\mathbf{b}$, solve $\mathbf{a}$: if $\mathbf{b}$ is given, the problem (4) turns into the following optimization problem:

$$\underset{\mathbf{a}}{\operatorname{argmin}}\ \|\mathbf{y}-[\mathbf{T},\mathcal{D}(\mathbf{W}_r\mathbf{b})]\,\mathbf{a}\|_2^2+\lambda\|\mathbf{a}\|_1,\quad \text{s.t. } \mathbf{a}_T\geq 0. \tag{6}$$

Let $\mathbf{P}=[\mathbf{T},\mathcal{D}(\mathbf{W}_r\mathbf{b})]$, $1_T\in R^m$ represents the column vector whose entries are all ones. Let $\psi(\mathbf{a})$ denote the indicator function defined by

$$\psi(\mathbf{a})=\begin{cases}0 & \boldsymbol{a}\geq 0\\ +\infty & \text{otherwise}\end{cases}. \tag{7}$$

Then, (6) can be optimized alternately as

$$\underset{\mathbf{a}}{\operatorname{argmin}}\ \|\mathbf{y}-\mathbf{P}\mathbf{a}\|_2^2+\lambda 1_T^{\mathrm{T}}\mathbf{a}_T+\lambda\|\tilde{\mathbf{a}}_I\|_{\mathbf{1}}+\psi(\mathbf{a}_T), \tag{8}$$

where the corner mark T represents transpose. We use the APG approach [3] to solve this minimization problem with

$$\begin{aligned}F(\mathbf{a})\ &=\|\mathbf{y}-\mathbf{P}\mathbf{a}\|_2^2+\lambda 1_T^{\mathrm{T}}\mathbf{a}_T\\ G(\mathbf{a})\ &=\lambda\|\tilde{\mathbf{a}}_I\|_1+\psi(\mathbf{a}_T)\end{aligned} \tag{9}$$

where $F(\mathbf{a})$ is a differentiable convex function and $G(\mathbf{a})$ is a nonsmooth convex function. In the APG algorithm, we need to solve an optimization problem:

$$a_{k+1}=\underset{\mathbf{a}}{\operatorname{argmin}}\ \frac{L}{2}\|\mathbf{a}-\beta_{k+1}+\nabla F(\beta_{k+1})/L\|_2^2+G(\mathbf{a}) \tag{10}$$

where L (in this letter, $L=20$) is the Lipschitz constant, k denotes the current iteration time, and β_{k+1} is defined in Algorithm 1. We define $g_{k+1}=\beta_{k+1}-\nabla F(\beta_{k+1})/L$ and the soft-thresholding operator: $S_\lambda(x)=\operatorname{sign}(x)\max(|x|-\lambda,\ 0)$.

Then, the algorithm for solving the minimization problem (6) is given in Algorithm 1.

The optimization problem in (3) can be iteratively solved by the above-mentioned two steps. The iteration operations are terminated when any of the following two conditions have been met: 1) the difference of objective values between two consecutive steps is smaller than a threshold (e.g., $\|J^i - J^{i-1}\|_2 \leq \varepsilon$, in this letter, ε is chosen as 0.01); 2) a maximal number Ω (in this work, $\Omega = 5$) of iterations has been met.

B. Constructing Weight Maps

In this work, we indicate that the weight maps should include two properties: 1) consistent motion property; and 2) forward–backward consistent tracking property. These two properties are discussed based on the pyramidal Lucas–Kanade tracker [14]. We track every pixels (if the target size is too large, we resize it to the maximal size of 40 × 40) of the target object using the method in [14]. Thus, we get a displacement set and we transform the displacements into the Euclidean distances, forming a distance set D.

Consistent motion property: The displacements of pixels which belong to the target are quite close to each other while the displacements of other pixels are random and disordered. Based on these observations, we use hierarchical agglomerative clustering [17] on D. Thus, D is partitioned into subset clusters $D^1, \ldots, D^s$, where s is the number of subsets. The number of elements in a subset can represent the reliabilities of the pixels in this subset.

Forward–backward consistent tracking property: The forward–backward error proposed in [18] can effectively identify the tracking reliability of each pixel. We apply it and get a forward–backward distance error set $E = \{e^1, \ldots, e^d\}$, where d is the number of pixels. The reliability value of the $i\text{th}$ pixel is then calculated as

$$r^i = \frac{1}{1 + \exp\left(-N\left(D^{o(i)}\right)/\sigma_1\right)} \frac{1}{\exp\left(e^i/\sigma_2\right)} \exp\left(-\frac{\left(x^i - x^c\right)^2}{2{\sigma_3}^2}\right), \tag{11}$$

where $N(X)$ represents the number of elements in set X, $o(i)$ indicates which subset cluster the $i\text{th}$ pixel belongs to. Variable x^i is the position of the $i\text{th}$ pixel and x^c is target center. Parameters σ_1 and σ_2 (in this work, $\sigma_1 = N/3$, $\sigma_2 = 15$) balance the contribution of the two measurements. The last term is a Gaussian function which indicates that pixels close to the target center are more reliable than those near the edges of target. Parameter σ_3 is proportional to the size of target object.

By (11), we obtain the reliability values of each pixel. Then, we choose the half-pixels which have the biggest half-reliability values as the reliable pixels. Then, the weights of these pixels are set to be a larger value k_1 (in this work, $k_1 = 200$) and the weights of other pixels are set to be a smaller value k_2 (in this work, $k_2 = 1$). Finally, we resize the weight map to the feature size, obtaining the weight map $\mathbf{w}$. Note that the constructed weight map preserves the reliable structure of the target. To make the model more discriminative, we use the inverse pattern of $\mathbf{w}$ as the background candidate weight map. This is especially useful for our model. The target candidate can be most possibly well-represented by the target templates together with the target weight maps and can be least possibly well-represented by the inverse patterns of them, whereas the two kinds of weight maps make no difference to background candidates.

Algorithm 2: The proposed tracker.

Input: The state set $\mathbf{X}_t = \{\mathbf{x}_t^i\}_{i=1}^M$, the candidate set $\mathbf{Y}_t = \{\mathbf{y}_t^i\}_{i=1}^M$, the template set $\mathbf{T}$ and weight map set $\mathbf{W}$. The initiation of $\mathbf{a}$.
1: **For** $i = 1, \ldots, N$
2: **For** $j = 1, 2, \ldots$, until converge or a maximal number Ω (in this work, $\Omega = 5$) of iterations have been met
3: Fix $\mathbf{a}_j^i$, obtain $\mathbf{b}_j^i$ by solving (5) using the LASSO method;
4: Fix $\mathbf{b}_j^i$, obtain $\mathbf{a}_{j+1}^i$ by solving (6) using Algorithm 1
5: **End**;
6: Obtaining the estimated parameters $\hat{\mathbf{b}}^i$ and $\hat{\mathbf{a}}^i$;
7: Computing the observation likelihood of each candidate $p^i = P(\mathbf{y}_t^i|\mathbf{x}_t^i)$ in (12);
8: **End**
Output:
9: Finding the $\mathbf{x}_t^*$ in (13);
10: Detecting the occlusion and updating the template set $\mathbf{T}$;
11: Updating the weight map set $\mathbf{W}$.

C. Tracking With the Proposed Algorithm

Our tracking method is based on the Bayesian filtering framework. Similar to [19], we use the affine motion model with six parameters to describe the object's state $\mathbf{x}_t = [l_x, l_y, \theta, s, r, \phi]^{\mathrm{T}}$, where $l_x, l_y, \theta, s, r, \phi$ denote x, y translations, rotation angle, scale, aspect ratio, and skew, respectively. The six affine parameters are modeled by six scalar Gaussian distributions independently. Then, we generate a candidate state set $\mathbf{X}_t = \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^M\}$. For each particle $\mathbf{x}_t^i$, we crop out the related image region as its feature $\mathbf{y}_t^i$. Then, we get a candidate set $\mathbf{Y}_t = [\mathbf{y}_t^1, \mathbf{y}_t^2, \ldots, \mathbf{y}_t^M] \in R^{N \times M}$.

The observation likelihood of state $\mathbf{x}_t^i$ is given as

$$P\left(\mathbf{y}_t^i \left| \mathbf{x}_t^i\right.\right) = \frac{1}{\varphi} \exp\left(\delta_1 \sum_{j=1}^{n} \left(b_j - b_{j+n}\right)\right) \times \exp\left(-\delta_2 \|\mathbf{y}_t^i - \mathbf{T}_t \mathbf{a}_T^i\|_2^2\right) \tag{12}$$

where φ is a normal factor and δ_1 and δ_2 (in this work, $\delta_1 = 1.5, \delta_2 = 0.1$) balance the contributions between the two terms. Coefficient $\mathbf{a}_T^i$ is the result of the optimization problem (3). Then, the optimal state $\mathbf{x}_t^*$ of frame *t* is obtained by

$$\mathbf{x}_t^* = \underset{\mathbf{x}_t^i \in \mathbf{X}_t}{\operatorname{argmax}} \, P\left(\mathbf{y}_t^i \left| \mathbf{x}_t^i\right.\right). \tag{13}$$

The detailed description of the proposed tracker is given in Algorithm 2. We update the target templates using the strategy proposed in [20] and use the newly obtained weight map to replace the oldest one in the weight map template set.

III. Experiments

The proposed method in this letter is implemented in MATLAB 2014a. We perform the experiments on a PC with Intel i7-4790 CPU (3.6 GHz) and 16 GB RAM memory and the tracker runs at 1.9 fps. We test the performance of the proposed tracker with 16 representative sequences and compare it with the top 12 state-of-the-art trackers, including SCM [21], ASLA [7], APGL1 [3], MTT [4], LSK [22], Struck [23], TLD

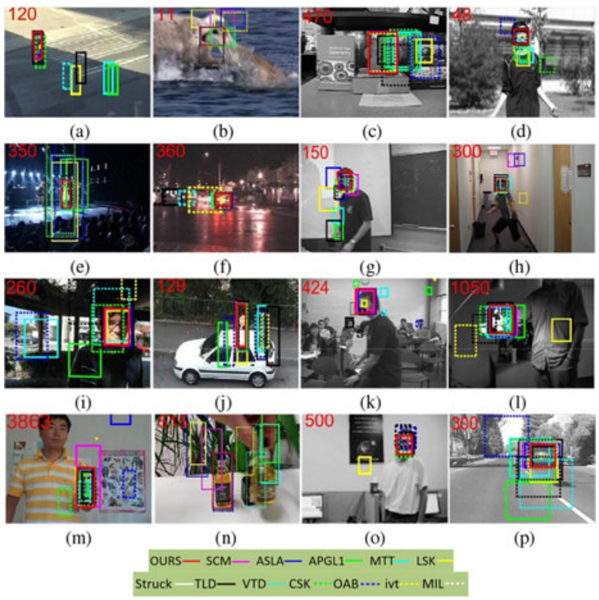

Fig. 2. Tracking results of the proposed method and the 12 state-of-the-art tracking algorithms on some representative frames of 16 sequences (*crossing*, *deer*, *fish*, *jumping*, *singer1*, *carDark*, *freeman1*, *boy*, *trellis*, *woman*, *freeman3*, *sylvester*, *doll*, *liquor*, *david2*, and *car4*, from left to right and top to bottom).

[24], VTD [25], CSK [26], OAB [27], IVT [19], and MIL [28]. We resize each observed sample to 24 × 24 pixels. The regularization factors λ and μ are set to be 0.02 and 1.2, respectively. The candidate number M in each frame is 600. Target template number m is adopted as 10.

Qualitative evaluation: The 16 sequences pose many challenging problems, including illumination variation [see Fig. 2(e), (f), (c), (i), (l), and (p)], scale variation [see Fig. 2(k), (g), and (m)], occlusion [see Fig. 2(e), (j), and (n)], deformation [see Fig. 2(a) and (j)], motion blur [see Fig. 2(b), (d), (h), and (n)], fast motion [see Fig. 2(b), (d), (h), and (j)], out-of-view [see Fig. 2(n)], background clutters [see Fig. 2(a), (b), (f), (i), and (n)], rotation [see Fig. 2(g), (h), (k), (l), (m), and (o)], and low resolution [see Fig. 2(k)].

Compared with the other five state-of-the-art sparse representation based trackers (e.g., SCM, ASLA, APGL1, MTT, and LSK), our tracker obviously performs better on sequences with occlusion and deformation, because it explores the reliable structure of target object while other trackers lose this information. The occluded pixels of target object can be represented by the trivial templates with less penalization. The reliable pixels of target object can be well-represented by the target templates themselves and thus they do not need to activate the corresponding trivial templates. If the trivial templates are activated to represent these reliable pixels (when the candidate is bad), they will be penalized more heavily. Fig. 2 shows the proposed tracker outperforms the 12 state-of-the-art methods in a wide range of challenging scenarios.

Quantitative evaluation: The proposed tracker and the 12 state-of-the-art trackers are evaluated by four criteria. VOR represents the well-known overlap rate. The CLE denotes the center location error in pixel. We also use the precision and success rate defined in [29] for evaluating.

Fig. 3 shows the precision and success plots illustrating the average distance precision and overlap precision over all the 16 sequences. In precision plots, the proposed method outperforms the well-known excellent trackers (Struck and SCM). This means that our tracker is more robust than the 12 state-of-the-art trackers in terms of the 16 challenging sequences. In success plots, the proposed method also performs the best. This means our tracker can handle scale variations and rotation well.

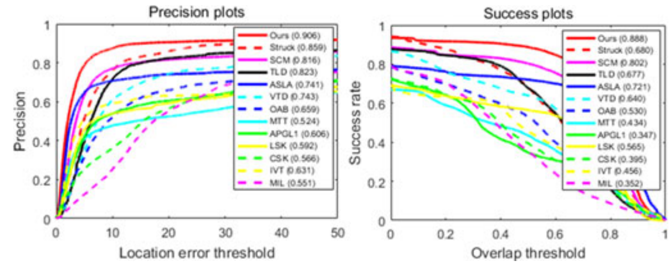

Fig. 3. Precision and success plots. The legend of the precision plot reports the average distance precision score at 20 pixels for each method and the legend of the success plot reports the success rate score with the threshold of 0.5.

TABLE I
RESULTS OF PRECISION (THRESHOLD = 20), SUCCESS RATE (THRESHOLD = 0.5), AVERAGE CENTER LOCATION ERROR (IN PIXEL), AND AVERAGE VOR OF OUR METHOD AND THE ELEVEN STATE-OF-THE-ART TRACKERS

	Precision (20)	SR(0.5)	Average CLE	Average VOR
Ours	**0.9059**	**0.8877**	**19.64**	**0.7338**
Struck	**0.8594**	0.6804	**19.22**	0.5846
SCM	0.8164	**0.8022**	**22.47**	**0.6594**
TLD	**0.8232**	0.6774	23.13	0.5578
ASLA	0.7409	**0.7209**	43.53	**0.6262**
VTD	0.7428	0.6395	26.96	0.5284
OAB	0.6589	0.5298	35.59	0.4545
LSK	0.5920	0.5649	56.43	0.4774
APGL1	0.6058	0.3466	56.38	0.3802
CSK	0.5663	0.3949	60.03	0.3973
MTT	0.5244	0.4338	132.1	0.3921
IVT	0.6306	0.4563	54.09	0.4231
MIL	0.5508	0.3525	46.93	0.3771

Table I demonstrates the statistical results of the trackers. Our tracker gains the average CLE of 19.64 pixels, which is the second smallest among all the compared trackers and very close to the best one. The success rate of our tracker is 0.8877, which exceeds the second best one by 0.0855 with the threshold of 0.5. The precision with the threshold of 20 and the average VOR of our tracker are 0.9059 and 0.7338, respectively. They are also the best ones among all the trackers and significantly outperform the others.

IV. CONCLUSION

This letter presents a novel sparse representation based tracking algorithm. In contrast to just use the target templates to sparsely represent the target, we propose to further constrain the model with a set of discriminative weight maps. The weight maps contain the reliable structures of the target object so that the occluded pixels of target object can be represented by the trivial templates with less penalization and if the reliable pixels are represented by the trivial templates, there will be more penalization. This strategy integrates the reliable structure information to traditional sparse representation, making the algorithm more robust. Qualitative and quantitative experimental results show that our approach outperforms many state-of-the-art methods.

References

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

[2] 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.

[3] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 1830–1837.

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

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

[6] D. Wang, H. Lu, and C. Bo, "Online visual tracking via two view sparse representation," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1031–1034, Sep. 2014.

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

[8] 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.

[9] 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.

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

[11] Z. Hong, X. Mei, D. Prokhorov, and D. Tao, "Tracking via robust multi-task multi-view joint sparse representation," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2013, pp. 649–656.

[12] T. Zhang *et al.*, "Structural sparse tracking," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2015, pp. 150–158.

[13] S. Zhang, H. Yao, H. Zhou, X. Sun, and S. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, vol. 100, no. 1, pp. 31–40, 2013.

[14] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, pp. 1–10, 2001.

[15] 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.

[16] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B*, vol. 67, pp. 267–288, 1996.

[17] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. Int. Conf. Pattern Recog.*, 2010, pp. 2756–2759.

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

[20] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient ℓ 1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2011, pp. 1257–1264.

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

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

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

[24] 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.

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

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

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

[28] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 983–990.

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