Visual Tracking via Coarse and Fine Structural Local Sparse Appearance Models

Xu Jia, Huchuan Lu, *Senior Member, IEEE*, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— Sparse representation has been successfully applied to visual tracking by finding the best candidate with a minimal reconstruction error using target templates. However, most sparse representation-based tracking methods only consider holistic rather than local appearance to discriminate between target and background regions, and hence may not perform well when target objects are heavily occluded. In this paper, we develop a simple yet robust tracking algorithm based on a coarse and fine structural local sparse appearance model. The proposed method exploits both partial and structural information of a target object based on sparse coding using the dictionary composed of patches from multiple target templates. The likelihood obtained by averaging and pooling operations exploits consistent appearance of object parts, thereby helping not only locate targets accurately but also handle partial occlusion. To update templates more accurately without introducing occluding regions, we introduce an occlusion detection scheme to account for pixels belonging to the target objects. The proposed method is evaluated on a large benchmark data set with three evaluation metrics. Experimental results demonstrate that the proposed tracking algorithm performs favorably against several state-of-the-art methods.**

***Index Terms*— Object tracking, coarse and fine structural local sparse appearance model, alignment-pooling.**

I. Introduction

VISUAL tracking has long been an important problem in computer vision including applications of video surveillance, vehicle navigation, motion analysis, and human computer interfaces, to name a few. While numerous tracking methods have been proposed, it remains a challenging problem due to factors such as partial occlusion, illumination change, pose change, background clutters and viewpoint variations.

In this paper, we propose an efficient tracking algorithm based on coarse and fine structural local sparse models. The proposed method exploits consistency of local appearance while the global appearance change over time. Image patches in a fixed spatial layout within a target region are extracted and encoded using a dictionary composed of patches from multiple target templates with sparsity constraints. The coding coefficients of patches across multiple templates are integrated via averaging and alignment pooling to obtain a robust representation of a target object. This operation helps locate an object accurately and handle partial occlusion effectively by exploring consistent parts of the target in an image sequence. To make its representation more distinctive and robust, we compute the likelihood of candidate regions based on the combination of patches extracted with coarse-and-fine strategy. The dictionary for local sparse coding is generated from the set of collected templates that are updated sequentially based on an incremental subspace learning method [1]. We introduce a scheme to detect occluding parts in roder to update template more accurately without including occluding pixels. The update module facilitates the proposed tracker account for target appearance variations caused by pose change and illumination change.

The contributions of this work are summarized as follows. First, sparse codes of local patches are computed via averaging and alignment pooling to model object appearance for visual tracking. A novel algorithm for constructing coarse and fine dictionaries is presented for robust representation. Second, a template update scheme based on incremental subspace learning is proposed to describe appearance change of objects. The template update module is equipped with an occlusion detection module to include pixels belonging to foreground objects. Third, extensive experiments on a large benchmark dataset are carried out to evaluate the performance of the proposed algorithm against the state-of-the-art methods.

II. Related Work and Problem Context

Broadly speaking, tracking algorithms can be categorized as either generative or discriminative. Discriminative methods formulate visual tracking as a classification problem which aims to distinguish foreground objects from background regions by exploiting visual information from both classes. Avidan [2], [3] uses support vector machines and boosting algorithms as classifiers respectively for visual tracking. Grabner and Bischof [4] propose an online boosting method to select discriminative features for dynamic scene change, and a semi-online boosting algorithm was proposed to address the drifting problem [5]. Babenko *et al*. [6] introduce an online multiple instance learning (MIL) approach for visual tracking, which puts ambiguous positive and negative samples into bags to learn a discriminative classifier. Kalal *et al*. [7] propose the

Manuscript received July 29, 2015; revised January 6, 2016 and April 26, 2016; accepted June 27, 2016. Date of publication July 18, 2016; date of current version August 5, 2016. The work of X. Jia and H. Lu was supported by the National Natural Science Foundation of China under Grant 61528101 and Grant 61472060. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianfei Cai.

X. Jia and H. Lu are with the Faculty of Electronic Information and Electrical Engineering, School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: jiayushenyang@gmail.com; lhchuan@dlut.edu.cn).

M.-H. Yang is with the Department of Electrical Engineering and Computer Science, 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.2016.2592701

P-N learning algorithm that exploits the underlying structure of positive and negative samples for learning effective classifiers for object tracking. Wang *et al.* [8] develop a discriminative appearance model based on superpixels which facilitates distinguish between target objects and background regions. As the structural appearance information is not fully exploited, these methods are likely to drift when the target objects undergo heavy occlusion and large scale change.

Generative methods formulate the tracking problem as searching for the region most similar to a target model. Templates based methods use color histogram [9] and pixel intensity [10] to model object appearance for visual tracking. However, simple representations do not work well when targets appear in cluttered backgrounds or heavily occluded as the spatial information of object appearance is not used. Matthews *et al.* [11] develop a template update method that alleviates the drifting problem by aligning with the example in the first frame. However, these approaches with single adaptive target appearance models are not likely to model the appearance variation well caused by large or lighting scale change, and heavy occlusion.

Numerous methods have been developed to address the above-mentioned issues. In [1], a subspace model is learned incrementally during the tracking process to account for appearance change of target objects. Kwon and Lee [12] utilize multiple observation and motion models within a particle filter framework to model appearance change caused by pose and illumination variation. Recently, tracking methods based on sparse representations have been proposed. Mei and Ling [13] and Mei *et al.* [14]) use a number of holistic templates of a target object as an appearance model and determine the most likely object regions by solving one ℓ_1 minimization problem for each drawn particle. Most of these methods employ holistic representation schemes and hence do not perform well when target objects are heavily occluded. Adam *et al.* [15] propose a fragment-based tracking method which partitions a target template into several patches where each one is tracked locally by measuring region similarity. The object location is estimated by combing the vote maps of these tracked patches. Liu *et al.* [16] employ a local appearance model based on histograms of sparse coefficients and the mean-shift algorithm for object tracking. However, most methods based on local appearance models assign equal weights to fragments and do not consider lage scale change.

Sparse representation has been successfully applied in numerous vision applications [13], [17]–[22]. With sparsity constraints, one image can be represented in the form of linear combination of only a few basis vectors. In [13] and [14], a target candidate is sparsely represented by a linear combination of the dictionary atoms constructed from target and trivial templates. However, this approach entails solving one ℓ_1 minimization with non-negativity constraints for each drawn particle for determining the most likely object location. The computational issues with ℓ_1 minimization problems are alleviated by approximation techniques and resampling methods [14], [23]. In [22], dynamic group sparsity constraints of spatial and temporal adjacency are introduced to model object appearance for robust tracking. Zhang *et al.* [24] propose

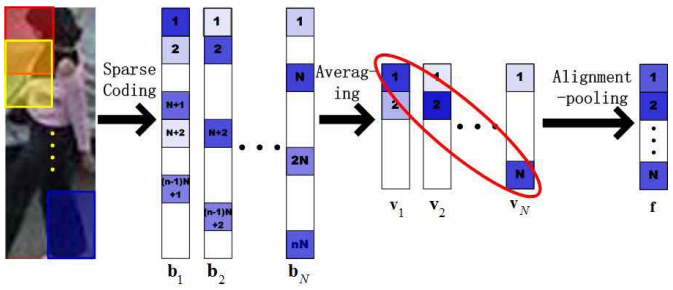

Fig. 1. Illustration of local patches and feature vectors formed by averaging and alignment pooling. Each local patch (where the first one is denoted in red, second one in yellow, and the last one in blue) is sparsely represented by the template set with a vector (where elements with larger values are denoted by darker color). These sparse coefficients are averaged and pooled to represent a target object.

a tracking method within the multi-task learning framework which exploits similarities among candidate regions based on joint group sparsity constraints. In [16] and [25], local sparse representation schemes are employed to model object appearance. The former method trains a classifier where an object is described by local sparse representation, and the latter one models the basis distribution of a target object with a histogram of sparse codes. Due to the representation of local patches, these methods perform well when target objects are heavily occluded. In addition, mean shift algorithms and voting maps are used to track target objects efficiently.

Our work bears some similarity to [16] in that both use sparse coding to model local appearance of the object. However, we extract coarse and fine local image patches in a fixed spatial layout to construct dictionaries. More importantly, the structural information contained in local patches are exploited to model target appearance. We show that histograms of local sparse coefficients can be integrated via the proposed averaging and alignment pooling to exploit consistent local appearance of objects for robust tracking. Instead of using fixed templates [15] or dictionary atoms [16] learned from the first frame, we update the proposed appearance model adaptively. In [13] and [14], the template is updated according to both the weights assigned to each template and the similarity between templates and current estimation of target candidate. In contrast, we use incremental subspace learning to update templates. Furthermore, an occlusion detection method is developed in this work to alleviate the problem where pixels belonging to the background regions are inadvertently included by straightforward model update schemes.

III. Coarse and Fine Structural Local Sparse Appearance Model

Most tracking methods use either multiple holistic templates or local appearance models to represent the target. In this paper, we develop coarse and fine structural local appearance models based on sparse representation of both multiple templates and local appearance models. That is, we exploit the consistent local appearance of the object and highlight the role of these fragments for visual tracking. Figure 1 shows the main steps of feature formation in this work.

A. Object-Specific Dictionary

For object classification and detection, a dictionary is usually learned by K-Means or sparse decomposition algorithm such as the K-SVD method [26]. A dictionary learned from patches or SIFT features [27] of a large dataset is expected to cover generic and distinctive patterns of objects from numerous classes. However, different from object classification and detection, object-specific dictionary models the target objects of interrest better for visual tracking. It has been shown that dictionaries composed of local image patches extracted from the region of a target object in a fixed spatial layout perform well for visual tracking [22], [25], [28].

To construct the dictionary, we collect a set of n templates to model object appearance $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_n]$ where $\mathbf{T}_i$ is an image observation of a target object. Given $\mathbf{T}$, we extract a set of local image patches inside the target region using a fixed spatial layout as shown in Figure 1. These local patches are used as the dictionary atoms to encode the local patches inside the possible candidate regions, i.e., $\mathbf{D} = \left[\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{(n\times N)}\right] \in \mathbb{R}^{d\times(n\times N)}$, where d is the dimension of the image patch vector, and N is the number of local patches sampled within the target region. Each column in $\mathbf{D}$ is obtained by ℓ_2 normalization on the vectorized local image patches extracted from $\mathbf{T}$. While each local patch represents one fixed part of the target object, the local patches altogether represent the holistic structure of the target. For a target candidate region, we extract the corresponding local patches with $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\right] \in \mathbb{R}^{d\times N}$.

B. Sparse Coding and Averaging

With the sparsity assumption, each local patch within a target region can be represented as the linear combination of only a few basis elements of the dictionary by solving

$$\begin{aligned} &\min_{\mathbf{b}_i} \ \|\mathbf{y}_i - \mathbf{D}\mathbf{b}_i\|_2^2 + \lambda\|\mathbf{b}_i\|_1, \\ &\text{s.t. } \mathbf{b}_i \succeq 0, \end{aligned} \tag{1}$$

where $\mathbf{y}_i$ denotes the i-th vectorized local image patch, $\mathbf{b}_i \in \mathbb{R}^{(n\times N)\times 1}$ is the corresponding sparse code of that local patch, and $\mathbf{b}_i \succeq 0$ enforces that all the elements of $\mathbf{b}_i$ are nonnegative. Note $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$ represents the sparse coefficients of patches within one candidate region. The proposed algorithm also accommodates another sparse coding method, that is, the elastic net method [29] based on ℓ_1/ℓ_2 regularization. Using the ℓ_1/ℓ_2 sparse coding method, each local patch is modeled by

$$\begin{aligned} &\min_{\mathbf{b}_i} \ \|\mathbf{y}_i - \mathbf{D}\mathbf{b}_i\|_2^2 + \lambda_1\|\mathbf{b}_i\|_1 + \frac{\lambda_2}{2}\|\mathbf{b}_i\|_2^2, \\ &\text{s.t. } \mathbf{b}_i \succeq 0. \end{aligned} \tag{2}$$

The elastic net method combines the ℓ_1 and ℓ_2 regularization together with the hope of getting the advantages of both. ℓ_1 regularization tends to find sparse solution but introduces a large Mean Square Error (MSE) error, while ℓ_2 is able to produce small MSE. The effects of these two coding methods on visual tracking are presented in Section VI.

Since the template set contains varying appearances of a target object, the local patterns that frequently appear at the same position are more distinctive than others, and play an important role for robust representation and good alignment. For example, the appearance change on the upper body of a pedestrian is much less than that of the lower body. Thus, it is more effective to recognize this person by the patches from the upper body than other parts. The encoding vector of a local patch is divided into several segments according to the template that each element of the vector corresponds to, i.e., $\mathbf{b}_i^\top = \left[\mathbf{b}_i^{(1)\top}, \mathbf{b}_i^{(2)\top}, \ldots, \mathbf{b}_i^{(n)\top}\right]$, where $\mathbf{b}_i^{(k)} \in \mathbb{R}^{N\times 1}$ denotes the segment of encoding vector $\mathbf{b}_i$ corresponding to the k-th template. To enhance the stability of the sparse coding results, we use equally weighted averaging on different segments of encoding vectors. That is, these segmented coefficients are equally weighted to obtain $\mathbf{v}_i$ for the i-th patch,

$$\mathbf{v}_i = \frac{1}{C}\sum_{k=1}^{n}\mathbf{b}_i^{(k)}, \quad i = 1, 2, \ldots, N, \tag{3}$$

where vector $\mathbf{v}_i$ corresponds to the i-th local patch and C is a normalization term. All the vectors $\mathbf{v}_i$ of local patches in a candidate region form a square matrix $\mathbf{V}$ and are further processed by the proposed pooling method.

C. Pooling

A single local patch can only capture some local appearance of the object. To model the whole object, it is necessary to pool the information contained in averaged coefficient vectors. We use the alignment pooling instead of the max pooling [18], [30], [31], or directly concatenating these vectors, as it helps alleviate the influence of irrelevant patches. In addition, the alignment pooling method makes full use of the structural information contained in the dictionary and hence helps to locate target objects accurately. After $\mathbf{v}_i$ is computed, each local patch at a certain position within a candidate region is represented by patches at different positions of templates. The local appearance of a patch with small variation is best described by the blocks at the same positions of the templates (i.e., using the sparse codes with the aligned positions). For example, the top left corner patch of the target object in Figure 1 should be best described by patches at the same position of templates. Therefore, the first element of $\mathbf{v}_1$ should have the largest coefficient value. We use the diagonal elements of the square matrix $\mathbf{V}$ as the pooled feature (See Figure 1), i.e.,

$$\mathbf{f} = \text{diag}(\mathbf{V}), \tag{4}$$

where $\mathbf{f}$ is the vector of proposed alignment pooled features. After consistent local patterns are computed by the equally weighted averaging operation, this pooling method further aligns local patterns between target candidate and templates based on the locations of structural blocks. Figure 2 shows that the tracking result that aligns well with the target object has a higher score computed with the proposed pooling method than the poorly aligned one.

The aligned tracking results also facilitate the incremental subspace learning for template update as the alignment pooling operation enables the proposed appearance model to deal with partial occlusion. When occlusion occurs, encodings of

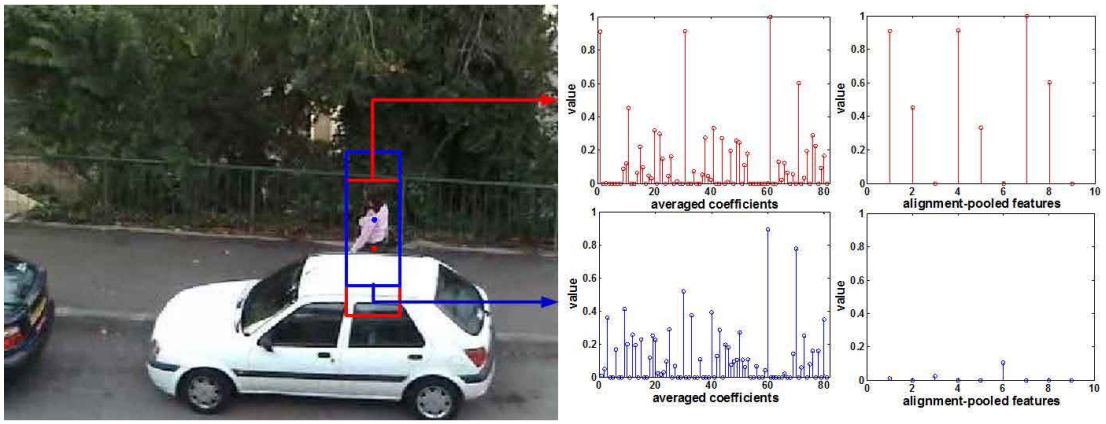

Fig. 2. Comparison of the pooled features obtained by averaging and alignment-pooling as for both good and bad candidates. The upper and lower rows show the pooled features for one good candidate (i.e., a region close to ground-truth tracking result) and one bad candidate (i.e., a region with large tracking error) regions. With only averaging operations, the sparse coefficients for two regions are similar. However, the sparse coefficients after alignment pooling are significantly different.

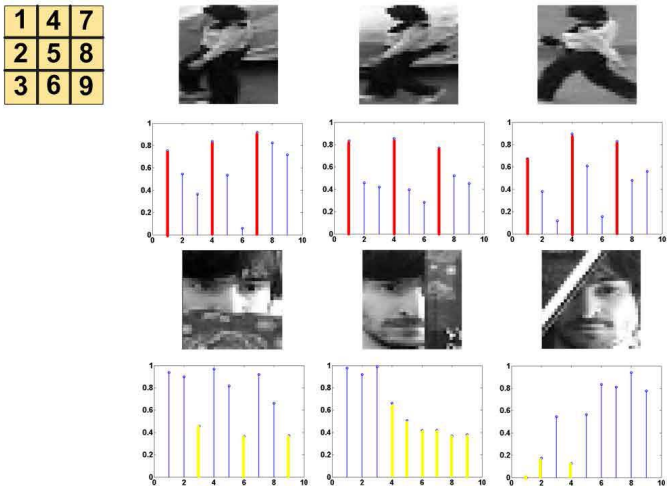

Fig. 3. Examples of pooled vectors. The proposed alignment pooling method facilitate determine stable parts either due to large appearance change (top) or partial occlusion (bottom) as indicated by the feature values.

the occluded local patches become dense due to significant appearance change, thereby leading to smaller values in $\mathbf{f}$. However, local patches which are not occluded have similar appearance to templates, and therefore can be described well by a few sparse coefficients and large values in $\mathbf{f}$. The similarity between target candidate regions and the set of target templates computed in this way is still higher than other candidates. Some examples of the pooled vectors are shown in Figure 3. In the first row, values correspond to the pooled vectors (denoted by red bars) of the upper body (patch 1, 4, and 7) are larger, and the tracking results align with the ground truth locations well. This shows the proposed alignment pooling method facilitates capturing stable structural parts of the target. In the second row, the proposed appearance model with alignment pooling helps handle partial occlusion by down-weighting the contribution of occluded local patches (denoted by yellow bars).

D. Coarse and Fine Representation

To further improve tracking performance, we adopt a coarse and fine representation to model target appearance. We extract local patches of different sizes within a target region and construct multiple dictionaries at coarse and fine scales. Based on these coarse and fine dictionaries, we obtain pooled features and compute coarse and fine similarities between candidate regions and a target model. The appearance model constructed at fine scale helps distinguish foreground targets from background regions whereas the appearance model at coarse scale facilitates account for appearance change due to large deformation. The final likelihood of a target region is computed by the combination of similarities computed at coarse and fine scales. We evaluate different weighting methods to compute the likelihood using Eq. 5 including similarities computed based on finer resolution patches are weighted more than those based on coarser ones; similarities computed based on coarser ones are weighted more; and they are equally weighted.

$$\eta_o = \sum_{m=1}^{M} \alpha^m \eta_o^m, \tag{5}$$

where η_o^m is the similarity computed on the m-th scale and α^m denotes the weight of the m-th scale. The effects of different weight combination are presented and discussed in Section VI.

IV. Template Update

Visual tracking with fixed templates is likely to fail in dynamic scenes as it does not consider inevitable appearance change due to factors such as illumination and pose variation. However, if the templates are updated too frequently with new observations, errors are likely to accumulate and the tracker will drift away from the target objects. Numerous approaches [1], [11], [13], [32] have been proposed for online update of appearance models. Ross *et al.* [1] extend the sequential Karhunen-Loeve algorithm and propose an incremental subspace learning algorithm to update both eigenbasis and mean vectors as new observations arrive. However the subspace based representation is sensitive to partial occlusion due to the assumption that the error term is Gaussian distributed with small variance. Mei and Ling [13] and Mei *et al.* [14] apply sparse representation to visual tracking and employ both target templates and trivial templates to

model object appearance and handle outliers (e.g., partial occlusion). However, this method is prone to fail when target objects are occluded by other objects with similar appearance. In this paper, we exploit both multiple templates and local sparse representation to adapt templates to appearance change of target objects, and reduce the influence of the occluded target templates.

When un-occluded, local patterns of a target object in the current and previous frames can be well represented mutually. However, it is not the case when a target object is occluded. Different from the appearance model described in Section III, we use patches of the tracking result to encode each patch $\mathbf{d}_i$ of each template in $\mathbf{T}$,

$$\min_{\mathbf{s}^{ij}} \left\| \mathbf{d}^{ij} - \hat{\mathbf{Y}}\mathbf{s}^{ij} \right\|_2^2 + \lambda \left\| \mathbf{s}^{ij} \right\|_1, \quad \text{s.t. } \mathbf{s}^{ij} \succeq 0, \tag{6}$$

where $\mathbf{d}^{ij}$ is the i-th patch of the j-th template $\mathbf{T_j}$, $\hat{\mathbf{Y}}$ represents local patches of the tracking result. Since the template set covers a wide range of target appearance, a non-occluded and consistent local pattern on the target is likely to match its corresponding patch at the same position of at least template. Hence, for a non-occluded patch of a target region, it is likely to use the one within the tracking result to represent the one within a template (i.e., only a few coefficients are large). However, an occluded patch may not match a similar one in the template set due to the drastic difference in appearance and thereby contributes little to the representation of the patch within a template (i.e., the coefficients are small). For each patch within the tracking result, we use the maximum of sparse coding coefficients over all templates. A threshold γ is used to determine whether one local patch within the tracking result can be used to effectively model target appearance in the previous frame.

$$\mathbf{t}_i = \max_j \{\mathbf{s}^{ij}\}, \quad i = 1, 2, \ldots, N, \quad j = 0, 1, \ldots, n-1, \tag{7}$$

where $\mathbf{t}_i$ is the maximum response corresponding to the i-th patch of the tracking results. We compute $\mathbf{g}_i$ to determine whether a local patch is occluded or not,

$$\mathbf{g}_i(r_i, c_i) = \begin{cases} 1, & \mathbf{t}_i(i) > \gamma, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where r_i and c_i are image coordinates of the pixel in the i-th patch $\mathbf{p}_i$. The occlusion of patches within the tracking result are accumulated to form a mask via Eq. 9 which can be used to describe the occlusion distribution within the whole target object.

$$\mathbf{M}(r, c) = \sum_{\{i | \mathbf{P}(r,c) = \mathbf{p}_i(r_i, c_i)\}} \mathbf{g}_i(r_i, c_i), \tag{9}$$

where $\mathbf{P}$ is the image patch of the tracking result, r and c are coordinates on that image patch.

In addition, we use the incremental subspace model [1] to represent a target object over a long duration. The tracking result is then reconstructed in terms of the occlusion extent of each local patch, and is represented as a weighted combination of its original appearance and its incremental subspace representation. This not only reduces the risk that the occlusion is updated into the dictionary, but adapts the model to appearance change of a target object. Finally, to reduce the artifacts of reconstruction by piecing together overlapped patches, the guided image filter [33] is used for smoothing. Hence, the new template is modeled as

$$\mathbf{T}_{new} = f(\mathbf{P} \odot \mathbf{M} + \mathbf{R} \odot (1 - \mathbf{M}), \mathbf{R}), \tag{10}$$

where f denotes the guided image filter, $\odot$ denotes element-wise multiplication, $\mathbf{T}_{new}$ represents the image patch of the new template, $\mathbf{R}$ is the reconstructed image patch using the incremental subspace reconstruction of the tracking result and also acts as guidance image. The new template $\mathbf{T}_{new}$ is then used for update of both the template set and the incremental subspace model. Some templates obtained from the above-mentioned process are shown in Figure 4.

(a) When a target object is un-occluded but has large pose variation.

(b) When a target object is occluded.

Fig. 4. Examples of the template set obtained by the proposed update method. (a) The template set consisted of images updated with different appearance change of the target object when it is not occluded. (b) The template set is updated with images without introducing occluding regions when the target object is occluded.

Algorithm 1 Template Update

Input: An image patch of the tracking result $\mathbf{P}$, a reconstructed image patch $\mathbf{R}$ based on an incremental subspace model, and a template set $\mathbf{T}$

1: Compute sparse codes for each local patch within the template according to Eq. 6;
2: Detect occluded patches according to Eq. 7 and Eq. 8;
3: Generate a mask for the tracked result according to the occlusion states of local patches using Eq. 9;
4: Obtain the new template $\mathbf{T}_{new}$ according to Eq. 10;
5: Add $\mathbf{T}_{new}$ to both the template set $\mathbf{T}$ and cumulative samples for incremental subspace

Output: New template set $\tilde{\mathbf{T}}$

The templates obtained when no occlusion occurs adapt to the appearance change of the target. When a target object is occluded, the templates focus on the parts which are not contaminated. With this template update approach, the proposed algorithm adapts to appearance change of target objects and handles occlusion as well. The template update method is summarized in Algorithm 1.

V. Bayesian Tracking Framework

Visual tracking is formulated within the Bayesian inference framework in this paper. Let affine parameters $\mathbf{x}_t = \{l_x, l_y, \theta, s, r, \tau\}$ represent the target state where l_x and l_y

denote the horizontal and vertical translation, θ denotes the rotation, s and r are the scale and aspect ratio, and τ is the skew parameter. Given a set of observations $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ up to the t-th frame, the target state variable $\mathbf{x}_t$ can be computed by the maximum a posteriori (MAP) estimation,

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t^i} p\left(\mathbf{x}_t^i | \mathbf{z}_{1:t}\right), \tag{11}$$

where $\mathbf{x}_t^i$ is the state of the i-th sample. Based on the Markov assumption, the posterior probability $p\left(\mathbf{x}_t^i | \mathbf{z}_{1:t}\right)$ can be inferred by the Bayes' theorem recursively,

$$p\left(\mathbf{x}_t^i | \mathbf{z}_{1:t}\right) \propto p\left(\mathbf{z}_t | \mathbf{x}_t^i\right) \int p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}\right) p\left(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}\right) d\mathbf{x}_{t-1}, \tag{12}$$

where $p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}\right)$ is the dynamic model and $p\left(\mathbf{z}_t | \mathbf{x}_t^i\right)$ denotes the observation model. The dynamic model $p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}\right)$ describes the temporal correlation of the target states between consecutive frames. The state variables are assumed to be independent of each other and Gaussian distribution is employed to model the target motion between two consecutive frames. The state transition is formulated as $p\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}\right) = N\left(\mathbf{x}_t^i; \mathbf{x}_{t-1}, \Sigma\right)$, where Σ is a diagonal covariance matrix whose elements are the variances of affine parameters. In this work, we only consider translationa and scale change, so we set the variance of affine parameters θ and τ to 0.

The observation model $p\left(\mathbf{z}_t | \mathbf{x}_t^i\right)$ describes the likelihood of the observation $\mathbf{z}_t$ at state $\mathbf{x}_t^i$, which plays an important role in robust tracking. The observation model in this work is characterized by the similarity between the candidates and the set of target templates,

$$p\left(\mathbf{z}_t | \mathbf{x}_t\right) \propto \eta_o, \tag{13}$$

where η_o represents the similarity described in Section III. With the template updated incrementally, the observation model is able to adapt to the appearance change well.

VI. Experiments

The proposed algorithm is implemented in MATLAB and runs at 2.5 frames per second on an Intel Core i7-860 (2.8 GHz) machine. The ℓ_1 and ℓ_1/ℓ_2 minimization problems are solved with the SPAMS package [34]. The regularization constant λ is set to 0.01, and λ_1 and λ_2 are respectively set to 0.05 and 0.01 in all experiments. For the dynamic model, we set the variances of the affine parameters to $\{l_x, l_y, \theta, s, r, \tau\} = \{6, 6, 0.01, 0.0, 0.005, 0\}$ for all experiments. The number of samples for particle filter is set to 800. For each sequence, the location of the target object is manually labeled in the first frame. We normalize each target image patch to 32×32 pixels and extract local patches of size 16×16 and 8×8 within the target region with 8 pixels as step length. The parameters n and N for local appearance models are set to 10 and 9 respectively. The threshold γ for occlusion detection is set to 0.5. As for the process of template update, 20 eigenvectors are used to carry out incremental subspace learning every 5 frames in all experiments.

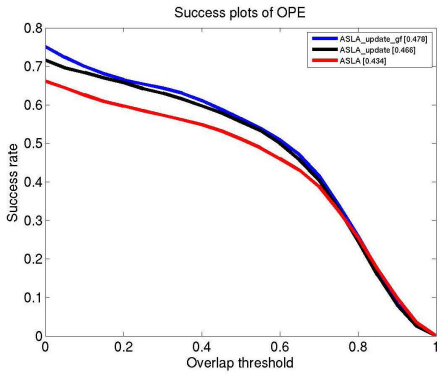

Fig. 5. Success plots of **OPE** for the different template update stratgies.

TABLE I
Proposed Algorithm With Different Configurations Based on Coarse and Fine Adaptive Structural Local Appearance Models

	$\alpha^f = 0.5, \alpha^c = 0.25$	$\alpha^f = 0.25, \alpha^c = 0.5$	$\alpha^f = \alpha^c = 0.5$
ℓ_1	MASLA_1	MASLA_2	MASLA_3
ℓ_1/ℓ_2	MASLA_4	MASLA_5	MASLA_6

We assess the proposed algorithm on a large benchmark dataset [35] using three evaluation criteria. The one-pass evaluation (**OPE**) criterion uses the ground truth object location in the first frame for evaluation. The spatial robustness evaluation (**SRE**) criterion perturbs the ground truth object location in the first frame with some offset and scale change for evaluation. The temporal robustness evaluation (**TRE**) initializes a tracker with ground truth objet locations at different frame for evaluation. For presentation clarity, only the top 10 trackers are presented in the following figures.

A. Comparision of Different Template Update Strategies

In this section, we present experimental results based on adaptive structural local sparse appearance (ASLA) [28] model but with different template update strategies. *ASLA_update* denotes the method using the proposed template update strategy without guided filter, and *ASLA_update_gf* denotes the proposed complete template update strategy. Figure 5 shows that the proposed template update strategy improves the tracking performance. That can be attributed to the added occlusion detection module. It alleviates the problem where pixels belonging to the background regions are inadvertently included by straightforward model update schemes. Besides, the use of guided filter also contributes to the improvement.

B. Proposed Algorithm With Different Configurations

In this section, we present experimental results using the proposed algorithm with different sparse coding methods and weights for coarse and fine representations. Table I shows different configurations of the proposed algorithm based on sparse coding methods (i.e., ℓ_1 and ℓ_1/ℓ_2) and weight combinations based on fine and coarse scales (α^f and α^c). We evaluate all six coarse and fine configurations and the tracking algorithm with ASLA on the benchmark dataset.

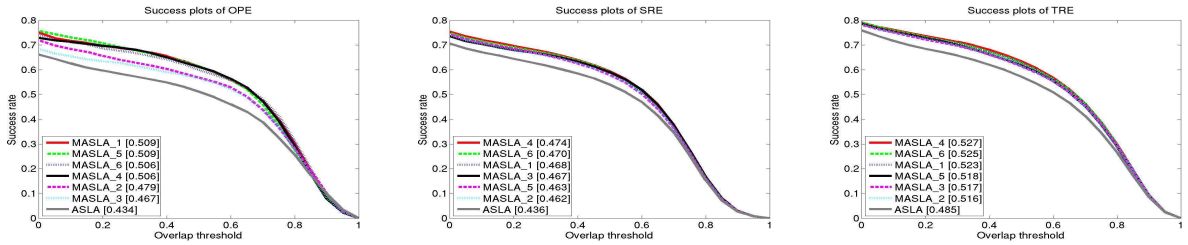

Fig. 6. Success plots of **OPE**, **SRE**, **TRE** for the proposed method with different configurations.

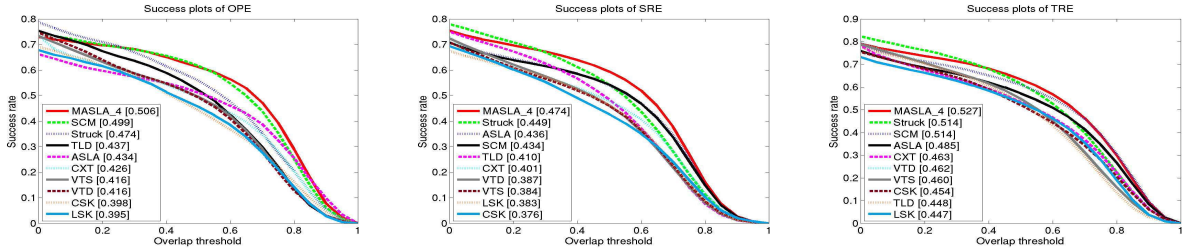

Fig. 7. Overall success plots of **OPE**, **SRE**, **TRE** for the proposed algorithm and state-of-the-art trackers.

Figure 6 shows the experimental results where all methods with the proposed coarse and fine adaptive structural local appearance (MASLA) models perform better than the single scale *ASLA* approach. These results indicate the effectiveness of coarse and fine local appearance models and the update module for visual tracking. Overall, the proposed algorithm with larger weights assigned to finer scale local appearance model performs better. This can be attributed to the fact that fine scale local appearance models capture more distinctive and representative patterns for separating foreground target objects from the background. Regarding sparse coding methods, the proposed algorithm with ℓ_1/ℓ_2 sparse coding (i.e., *MASLA_4*) performs slightly better due to its stability.

C. Comparison With State-of-the-Art Methods

In this section, we evaluate the proposed algorithm against 29 state-of-the-art tracking methods on the benchmark dataset using three evaluation metrics with success plots. We use the *MASLA_4* tracking method for comparisons with other state-of-the-art approaches. Figure 7 shows the overall performance of the evaluated tracking methods. The proposed *MASLA_4* algorithm performs favorably against the other trackers using all the evaluation metrics, especially **SRE** and **TRE** that are designed to evaluate robustness with respect to spatial and temporal perturbations. Through the operations of averaging and alignment pooling, the proposed algorithm exploits consistent local patterns for robust visual tracking, and reduces the effects of noisy occluding pixels. The proposed coarse and fine structural representation further enhances robustness to scales of initialization. We further analyze the tracking results based on challenging attributes of the sequences in the benchmark dataset.

1) Occlusion: Figure 8 shows the top performing tracking methods for image sequences where target objects are occluded where the *SCM* [36] and *MASLA_4* methods perform well. We note that these two methods use local appearance models based on sparse representation. The averaging and alignment pooling operators of the proposed algorithm exploit consistent and un-occluded local patterns of the object for visual tracking. In addition, the influence of occluded pixels is alleviated by the proposed pooling method (See Figure 2). The template update module facilitates occluding pixels from being included in the appearance model (See Figure 4) as occluders are detected by comparing local patches within the estimated candidate region and the corresponding ones in the template set.

2) Illumination Variation: For sequences where objects undergo large illumination change, local appearance based methods such as *MASLA_4*, *SCM*, *ASLA* and *LSK* [16] perform as shown in Figure 9. Although illumination variation is not uniform on a target object (i.e., holistic view), the intensity change on local patches tend to be the same. The ℓ_2 normalization of local image regions and the sparse representation facilitate the proposed local appearance model to account for illumination change of target objects. For the *Struck* [37] and *TLD* [7] methods that perform well in these image sequences, local contrast features such as Haar-like features and pixel comparisons are used to represent target objects.

3) Deformation: Similar to most existing online tracking methods, the proposed algorithm is developed mainly for rigid objects with certain deformation. The appearance change due to deformation is accounted for by the averaging and pooling operators. The local patches with frequently changing appearance do not always have good matches at the same position of the templates. That is, it requires a few dictionary words to describe these patches. As such, the weights of such patches are lower after averaging and alignment pooling (Similar to the example illustrated in Figure 2). Overall, the proposed *MASLA_4* method performs well in handling object deformation as shown in Figure 10.

4) Background Clutters: For image sequences with multiple similar objects, tracking methods based on generative

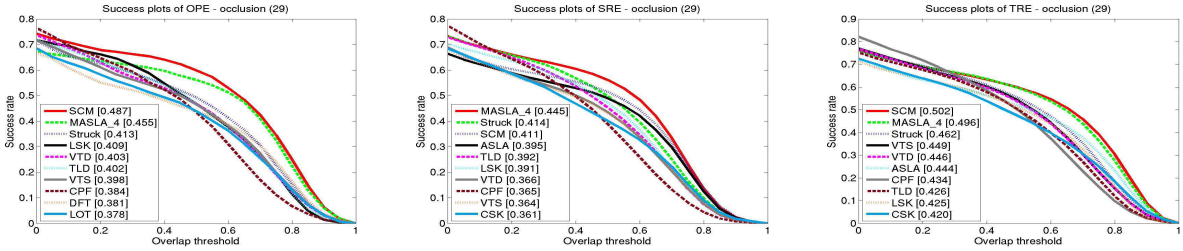

Fig. 8. Success plots of **OPE**, **SRE**, **TRE** in sequences with occlusion.

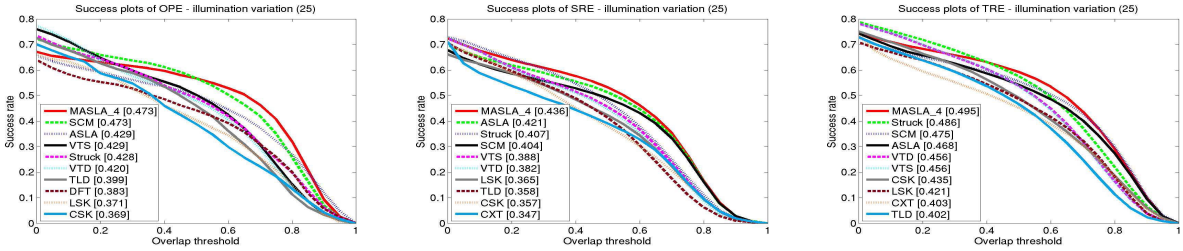

Fig. 9. Success plots of **OPE**, **SRE**, **TRE** in sequences with illumination variation.

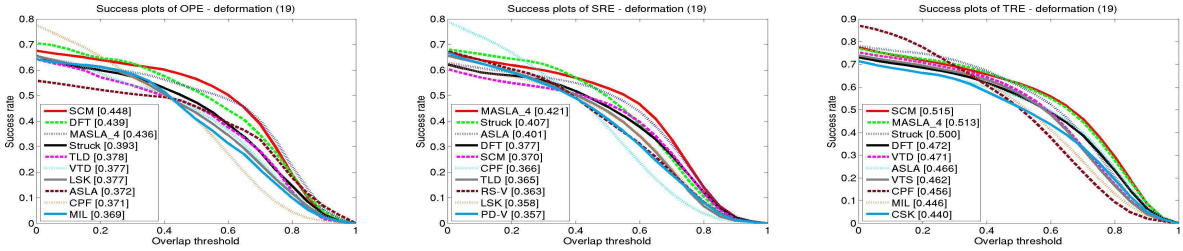

Fig. 10. Success plots of **OPE**, **SRE**, **TRE** in sequences with deformation.

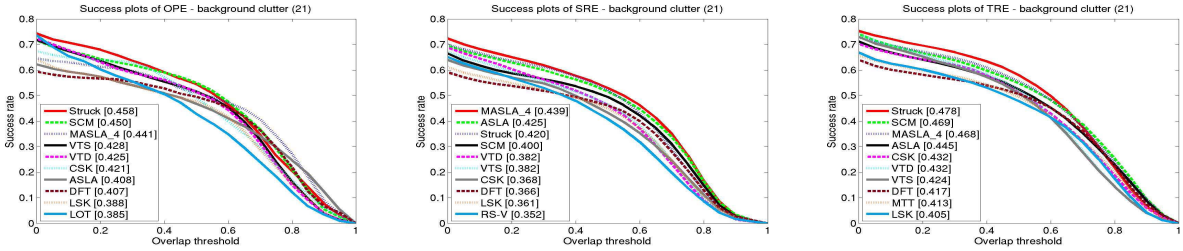

Fig. 11. Success plots of **OPE**, **SRE**, **TRE** in sequences with cluttered background.

representations may not perform well. On the other hand, discriminative methods such as *Struck*, *SCM*, *MASLA_4*, and *ASLA*, perform favorably as shown in Figure 11. These discriminative methods construct appearance models based on a template set covering a wide range of target appearance for separating foreground objects from the background. Moreover, the coarse and fine strategy in the proposed method allows it to explore distinctive and representative local patterns of different scales, thereby reducing the possibility of drifting to other objects or background. In addition, generative approaches such as *VTD* [12] and *VTS* [38] are also effective in dealing with cluttered background by using multiple representations.

5) Scale Variations: We note that visual tracking methods typically do not perform well when objects undergo large scale change (e.g., *David* and *singer* sequences). For image sequences with scale variations, the *MASLA_4*, *SCM* and *ASLA* methods perform well as shown in Figure 12. All these methods use affine motion models to account for large appearance variation instead of translation or similarity transforms.

6) Fast Motion: Figure 13 shows the tracking results with image sequences containing fast moving objects. Overall, the *Struck*, *TLD*, *CXT* [39] and *OAB* [4] methods perform well. As a large number of candidate regions are generated by dense sampling over a large range, these methods are effective in dealing with fast moving objects. On the other hand, the *MASLA_4*, *SCM* and *IVT* [1] methods do not perform well as the number of particles are fixed in all experiments as a trade-off between speed and accuracy. We note that the tracking methods based on particle filters are likely to perform better by drawing more particles.

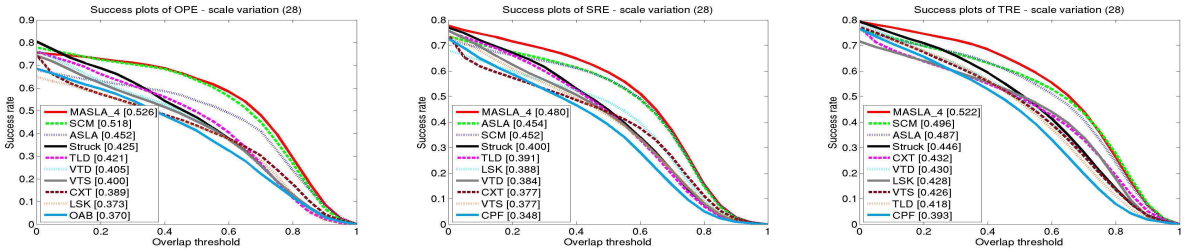

Fig. 12. Success plots of **OPE**, **SRE**, **TRE** in sequences with scale variation.

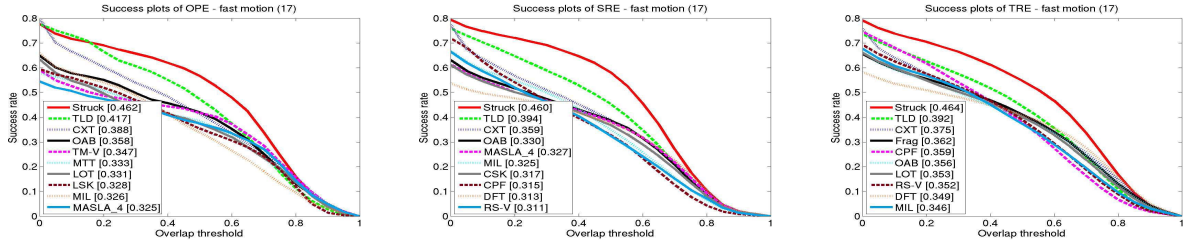

Fig. 13. Success plots of **OPE**, **SRE**, **TRE** in sequences with abrupt motion.

VII. Conclusion

In this paper, we propose an efficient tracking algorithm based on coarse and fine structural local sparse appearance models with adaptive update. The proposed algorithms exploit both structural and local information of target objects by averaging and alignment pooling. By using consistent and distinct local object appearance, the proposed algorithms are able to track targets more accurately and robustly under occlusion and clutters. The update scheme based on occlusion detection alleviates the problem where incorrectly estimated or occluding pixels are included in the template set during the update process. Experimental results with comparisons to numerous state-of-the-art methods on a large benchmark dataset demonstrate the effectiveness and robustness of the proposed algorithms.

References

[1] 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.

[2] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[3] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[4] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 260–267.

[5] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

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

[7] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 49–56.

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

[9] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.

[10] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

[11] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

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

[13] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1436–1443.

[14] 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. Vis. Pattern Recognit.*, Jun. 2011, pp. 1257–1264.

[15] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 798–805.

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

[17] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

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

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

[20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[21] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 625–632.

[22] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 624–637.

[23] 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.

[24] 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.

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

[26] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 228–247, 2008.

[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[28] 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.
[29] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc., B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[30] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, pp. 1019–1025, Nov. 1999.
[31] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.
[32] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2001, pp. 415–422.
[33] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. 11th Eur. Conf. Comput. Vis.*, vol. 1. 2010, pp. 1–14.
[34] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.
[35] 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.
[36] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.
[37] 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.
[38] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1195–1202.
[39] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1177–1184.

Xu Jia received the B.E. degree in electronic information engineering and the M.S. degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with Katholieke Universiteit Leuven, Belgium. His research interests include visual tracking, vision and language, and deep learning.

Huchuan Lu (SM'12) received the M.Sc. degree in signal and information processing and the Ph.D. degree in system engineering from the Dalian University of Technology (DUT), China, in 1998 and 2008, respectively. He has been a Faculty Member since 1998 and a Professor since 2012 with the School of Information and Communication Engineering, DUT. His research interests are in the areas of computer vision and pattern recognition. In recent years, he focuses on visual tracking and segmentation. Now, he serves as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS.

Ming-Hsuan Yang (SM'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Urbana, in 2000. He was a Senior Research Scientist with the Honda Research Institute, working on vision problems related to humanoid robots. He is currently an Assistant Professor with the Department of Electrical Engineering and Computer Science, University of California (UC) at Merced. He has co-authored the book entitled *Face Detection and Gesture Recognition for Human-Computer Interaction* (Kluwer, 2001). He edited the Special Issue on Face Recognition of *Computer Vision and Image Understanding* in 2003. He is currently a Senior Member of the Association for Computing Machinery. He was a recipient of the Ray Ozzie Fellowship for his research work in 1999. He received the Natural Science Foundation CAREER Award in 2012, the Campus Wide Senate Award for Distinguished Early Career Research at UC in 2011, and the Google Faculty Award in 2009. He edited a Special Issue on Real World Face Recognition for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He served as an Area Chair for the IEEE International Conference on Computer Vision in 2011, the IEEE Conference on Computer Vision and Pattern Recognition in 2008 and 2009, and the Asian Conference on Computer in 2009, 2010, and 2012. He served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and the *Image and Vision Computing*.