Inverse Sparse Tracker With a Locally Weighted Distance Metric

Dong Wang, Huchuan Lu, *Senior Member, IEEE*, Ziyang Xiao, and Ming-Hsuan Yang, *Senior Member, IEEE*

***Abstract*— Sparse representation has been recently extensively studied for visual tracking and generally facilitates more accurate tracking results than classic methods. In this paper, we propose a sparsity-based tracking algorithm that is featured with two components: 1) an inverse sparse representation formulation and 2) a locally weighted distance metric. In the inverse sparse representation formulation, the target template is reconstructed with particles, which enables the tracker to compute the weights of all particles by solving only one ℓ_1 optimization problem and thereby provides a quite efficient model. This is in direct contrast to most previous sparse trackers that entail solving one optimization problem for each particle. However, we notice that this formulation with normal Euclidean distance metric is sensitive to partial noise like occlusion and illumination changes. To this end, we design a locally weighted distance metric to replace the Euclidean one. Similar ideas of using local features appear in other works, but only being supported by popular assumptions like local models could handle partial noise better than holistic models, without any solid theoretical analysis. In this paper, we attempt to explicitly explain it from a mathematical view. On that basis, we further propose a method to assign local weights by exploiting the temporal and spatial continuity. In the proposed method, appearance changes caused by partial occlusion and shape deformation are carefully considered, thereby facilitating accurate similarity measurement and model update. The experimental validation is conducted from two aspects: 1) self validation on key components and 2) comparison with other state-of-the-art algorithms. Results over 15 challenging sequences show that the proposed tracking algorithm performs favorably against the existing sparsity-based trackers and the other state-of-the-art methods.**

Manuscript received September 2, 2014; revised April 3, 2015; accepted April 12, 2015. Date of publication April 28, 2015; date of current version May 19, 2015. This work was supported in part by the China Post-Doctoral Science Foundation under Grant 2014M551085, in part by the Natural Science Foundation of China under Grant 61472060, in part by the Fundamental Research Funds for the Central Universities under Grant DUT13RC(3)105 and Grant DUT14YQ101, in part by the China Mobile Communication Corporation under Grant MCM20122071, and in part by the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing, China, under Grant 30920140122007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aydin Alatan.

D. Wang is with the School of Information and Communication Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China, the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China, and also with the Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wdice@dlut.edu.cn).

H. Lu and Z. Xiao are with the School of Information and Communication Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: lhchuan@dlut.edu.cn; xiaoziyang1028@gmail.com).

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.2015.2427518

***Index Terms*— Visual tracking; sparse representation; inverse sparse tracker; robust distance.**

I. Introduction

OBJECT tracking is of great importance for numerous computer vision applications including video surveillance, motion analysis, robotics, human computer interaction, to name a few. While numerous tracking algorithms have been proposed in the past decades (like [1], [3], [4], [6]–[8], [11], [13]–[16], [25], [26], [32], and [33], etc), it remains a challenging task due to large target appearance variations caused by numerous challenging factors including heavy occlusions, illumination change, pose variation, shape deformation, motion blur and background clutter (see Figure 1).

Sparse representation has been applied to numerous computer vision tasks including object tracking [5], [12], [19], [22], [27], [35]–[39], image super-resolution [30], [31], image restoration [21], object detection [2], etc. It is usually posed as an ℓ_1 minimization problem and various algorithms have been proposed [28]. Based on a generative formulation for object tracking, Mei and Ling [22] present a method that reconstructs each candidate region (generated by a particle filter scheme) with dictionary atoms composed of target and trivial templates. The corresponding sparse coefficient vector of each candidate is computed by solving one ℓ_1 minimization problem with non-negativity constraints. In [19], Liu *et al.* present a tracking method to select a sparse and discriminative set of features for representing the tracked objects. Wang *et al.* [27] learn sparse codes of local image patches and train a linear classifier to discriminate the target from the background for object tracking. In [18], Liu *et al.* learn a compact dictionary from local patches and use the sparse coefficients within the mean shift framework for achieve robust tracking. Jia *et al.* [12] propose a structural local sparse appearance model, which integrates local and global information through a pooling method. To combine the strength of both generative and discriminative ways, Zhong *et al.* [36] develop a sparsity-based collaborative model for object tracking. The four works ([12], [17], [27], [36]) all utilize local features in a sparse representation formulation as a technique to overcome partial occlusion. In this work, we provide a novel mathematical analysis on this popular technique so that we obtain a more clear and substantial understanding of its advantage and naturally find a way to benefit more from it.

Except [17] which seeks mode shifts with the mean shift algorithm, all above-mentioned approaches solve one ℓ_1 minimization problem for each sample drawn by particle filter. The computational load increases significantly

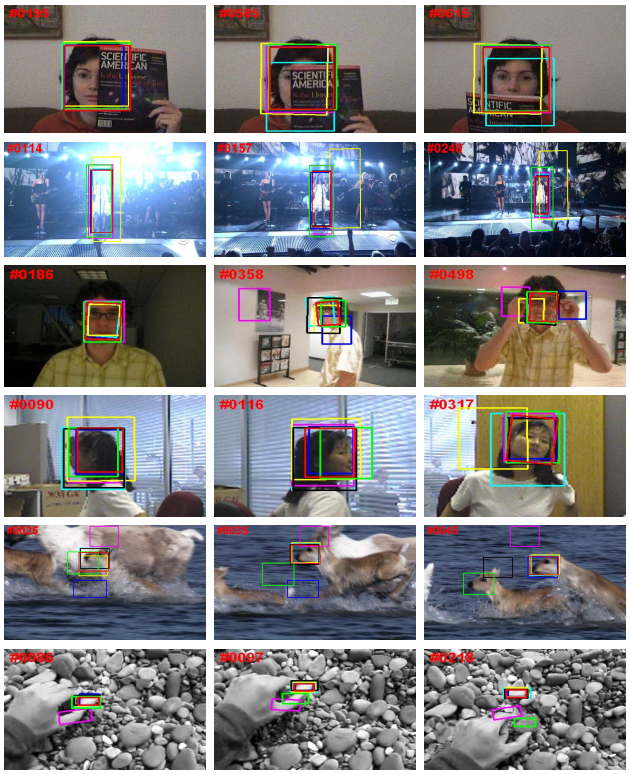


Fig. 1. Challenges of visual tracking in real-world environments, including heavy occlusions (*Face1*), illumination change (*Singer1*), shape deformation (*David1*), in-and-out plane rotation (*Girl*), motion blur (*Deer*) and background clutter (*Stone*). We use cyan, blue, yellow, magenta, green, black and red rectangles to represent the tracking results of the ASLSA [12], L1APG [5], MTT [35], ODLT [27], SPT [18], SCM [36] and the proposed method, respectively.

when the number of particles is large, thereby limiting their applicability for real-time tasks. Efforts have been made to reduce the time complexity for sparsity-based tracking algorithms. In [23], an approximate solution is developed to reduce the number of particles for object tracking. Using a different gradient descent approach, the computational load of the ℓ_1 tracker is further reduced [5]. In [16], Li *et al.* propose a tracking algorithm which exploits the restricted isometry property in compressive sensing to reduce the dictionary dimension and apply an orthogonal matching pursuit algorithm to solve ℓ_0 minimization problems. Liu and Sun [20] propose a tracking method by using a dictionary composed by all candidates and trivial templates to represent a static object template by solving one ℓ_1 minimization problem. Zhang *et al.* [35] formulate the tracking problem based on sparse representation within the multi-task learning framework in which the similarities between candidates are exploited by enforcing joint group sparsity. Similar approaches are presented in [34], in which the particle-based tracking problem is posed as a low-rank matrix learning problem. However, although good candidates are likely to resemble each other, the appearances of bad candidates are diverse. The dissimilarities can be reflected by the diversity of candidate sparse coefficients which, however, is diminished by mixed norm constraints that encourage coefficients to be similar with each other.

In this work, we propose a fast and effective tracking algorithm based on an inverse sparse representation formulation with a locally weighted distance metric. In most previous sparse tracking formulations (such as [5], [18], [22], [35], and [36]), all candidates have to be reconstructed by target templates with the sparsity constraints and therefore hundreds of ℓ_1 optimization problems need to be solved per frame. In contrast to them, we formulate object tracking as solving only one sparse representation problem per frame, and therefore denote the proposed method as an inverse sparse formulation. The main purpose of our method is to select a subset of weighted candidates to represent the target template and determine the optimum tracking result. This formulation is able to improve computational efficiency significantly. However, with the normal Euclidean distance metric, it is sensitive to partial noise like occlusion and illumination changes. Therefore, we design a robust locally weighted distance metric. The issues with the Euclidean distance metric are addressed thoroughly with a mathematical analysis, which eventually leads us to the new metric. Then, we assign the local weights by exploiting the temporal and spatial continuity in the meanwhile. In this scheme, the appearance changes caused by partial occlusion and shape deformation are carefully considered. With this metric, the local parts that are not occluded or remain unchanged with moderate shape deformation are weighted more than the others such that appearance change of the target can be better accounted for and the distance between the target template and each candidate can be measured more accurately. Finally, by using fifteen challenging video clips, we evaluate the proposed tracking algorithm with twelve state-of-the-art trackers as well as some reference algorithms designed for self comparison. The results demonstrate that our tracker performs favorably in terms of both effectiveness and efficiency.

II. The Proposed Algorithm

A. Locally Weighted Distance Metric

In this subsection, we explore the reason why the Euclidean metric loses its accuracy in the existence of partial impulse noise by locally decomposing and analyzing it. The obtained conclusion naturally leads us to the way of designing a more robust metric, i.e. the one we named as the locally weighted distance metric.

1) Locally Decomposed Euclidean Distance Metric: The error of a template $\mathbf{t} \in \mathbb{R}^{d\times 1}$ reconstructed by a candidate $\mathbf{y} \in \mathbb{R}^{d\times 1}$ can be represented by their squared Euclidean distance as

$$\left\|\bar{\mathbf{t}} - \bar{\mathbf{y}}\right\|^2 = \left\| \frac{\mathbf{t}}{\|\mathbf{t}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\|^2, \tag{1}$$

where $\bar{\mathbf{y}} = \frac{\mathbf{y}}{\|\mathbf{y}\|}$ represents the candidate feature vector with unit ℓ_2 norm.

We analyze the Euclidean metric from a local perspective. For the candidate $\mathbf{y} \in \mathbb{R}^{d\times 1}$, we reorganize it as the concatenation of N local feature vectors $\mathbf{y} = \left[\mathbf{y}_1^\top, \mathbf{y}_2^\top, \ldots, \mathbf{y}_N^\top\right]^\top$, where $\mathbf{y}_i \in \mathbb{R}^{l\times 1}$ is a column vector denoting the i-th local part and $N = d/l$ represents the number of local parts. Likewise, $\mathbf{t}$ can be represented in the same way. Then equation (1) can

be expanded as

$$\begin{aligned}\left\|\bar{\mathbf{t}}-\bar{\mathbf{y}}\right\|^2 &= 2-2\frac{\langle \mathbf{t},\mathbf{y}\rangle}{\|\mathbf{t}\|\,\|\mathbf{y}\|} = 2-2\sum_{i=1}^{N}\frac{\langle \mathbf{t}_i,\mathbf{y}_i\rangle}{\|\mathbf{t}\|\,\|\mathbf{y}\|}\\ &= 2-2\sum_{i=1}^{N}\frac{\|\mathbf{t}_i\|\,\|\mathbf{y}_i\|}{\|\mathbf{t}\|\,\|\mathbf{y}\|}\frac{\langle \mathbf{t}_i,\mathbf{y}_i\rangle}{\|\mathbf{t}_i\|\,\|\mathbf{y}_i\|}\\ &= 2-2\sum_{i=1}^{N}(W_i\times\rho(\mathbf{t}_i,\mathbf{y}_i)),\end{aligned} \tag{2}$$

where $\rho(\mathbf{t}_i,\mathbf{y}_i)=\langle \mathbf{t}_i,\mathbf{y}_i\rangle/(\|\mathbf{t}_i\|\,\|\mathbf{y}_i\|)$ clearly indicates the local similarity between $\mathbf{t}_i$ and $\mathbf{y}_i$ since it is actually the cosine value of the two vectors, then the part W_i could be interpreted as the weight for the local similarity $\rho(\mathbf{t}_i,\mathbf{y}_i)$.

However, the rightfulness of the local weight is in serious doubt. If we further decompose the weight as

$$W_i=\frac{\|\mathbf{t}_i\|\,\|\mathbf{y}_i\|}{\|\mathbf{t}\|\,\|\mathbf{y}\|}=R_i^{\mathbf{t}}\times R_i^{\mathbf{y}},\quad i=1,\ldots,N, \tag{3}$$

where $R_i^{\mathbf{t}}=\frac{\|\mathbf{t}_i\|}{\|\mathbf{t}\|}$ and $R_i^{\mathbf{y}}=\frac{\|\mathbf{y}_i\|}{\|\mathbf{y}\|}$. Then it turns out that the weights are totally determined by the values of $R_i^{\mathbf{t}}$ and $R_i^{\mathbf{y}}$. Considering that in most sparse representation formulations pixel intensities are utilized as features, $R_i^{\mathbf{t}}$ and $R_i^{\mathbf{y}}$ actually represent the brightness ratios of the i-th image parts $\mathbf{t}_i$ and $\mathbf{y}_i$ to the global image vectors $\mathbf{t}$ and $\mathbf{y}$. That is to say, the Euclidean distance metric would intrinsically assign heavier weights to the brighter local parts, which, however, is not advisable since a brighter part does not mean it is more significant or discriminative.

More importantly, if some impulse noise (like partial occlusions) occurs to a few local parts of a candidate $\mathbf{y}$, all the local brightness ratios $\{R_i^{\mathbf{y}}\}_{i=1}^N$ would change significantly. These changes would be highly random and unpredictable since it is impossible to well predict the intensities of the impulse noises. Consequently, the local similarities weights $\{W_i\}_{i=1}^N$ also change with high randomness, which makes the Euclidean distance metric less robust in such cases.

The first four rows in Figure 2 show an example for this situation. When impulse noise (like partial occlusion in $\mathbf{Y^2}$ and $\mathbf{Y^3}$) occurs, the local weights are highly dependent on the intensities of noises and become not credible any more. In the second and third rows of Figure 2, the local similarities between the template $\mathbf{T}$ and $\mathbf{Y^2}$ are the same with that between $\mathbf{T}$ and $\mathbf{Y^3}$ while their local weights in Euclidean distance metric are quite different. In $\mathbf{Y^2}$, the intensities between the black local occlusion and its corresponding original patch (in $\mathbf{Y^1}$) are very similar, which makes that the overall weight distribution does not change too much in this case. Thus, the final Euclidean distance between $\mathbf{T}$ and $\mathbf{Y^2}$ happens to be reasonable. However, a gray partial corruption in the very same local regions in $\mathbf{Y^3}$ brings adverse effects. In $\mathbf{Y^3}$, the intensity difference between the gray local occlusion and its corresponding original patch (in $\mathbf{Y^1}$) is very large, which changes the original weight distribution significantly and further makes the Euclidean distance even larger than that of the template $\mathbf{T}$ and the worse candidate $\mathbf{Y^4}$. Consequently, tracking algorithms based on such information are likely to drift from the tracked object gradually.

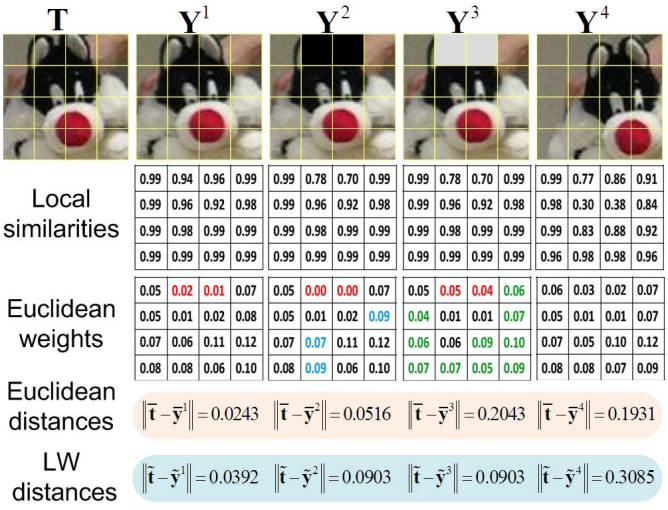


Fig. 2. $\mathbf{T}$ is a template; $\mathbf{Y^1}$ is a good candidate while $\mathbf{Y^2}$ and $\mathbf{Y^3}$ are extracted from the same candidate state with $\mathbf{Y^1}$, but being occluded in local regions (upper center part); $\mathbf{Y^4}$ is a bad candidate. The second row displays the local similarities between the template and candidates while the third row demonstrates the weights for local similarities assigned by Euclidean distance. The fourth and the last row respectively displays the Euclidean distances and the locally weighted distances among the template and candidates.

2) Locally Bounded Distance Metric: To design a more robust metric without any prior knowledge, a straightforward way is to assign uniform weights for local similarities, i.e., $W_i=\frac{1}{N},\quad i=1,\ldots,N$. Then, we represent the distance in this new metric as $d^{\ell}(\mathbf{t},\mathbf{y})$:

$$d^{\ell}(\mathbf{t},\mathbf{y})=2-2\sum_{i=1}^{N}\frac{1}{N}\rho(\mathbf{t}_i,\mathbf{y}_i)=\frac{1}{N}\sum_{i=1}^{N}\left\|\frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}-\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|}\right\|^2. \tag{4}$$

Let

$$\hat{\mathbf{y}}=[\frac{\mathbf{y}_1^{\top}}{\|\mathbf{y}_1\|},\frac{\mathbf{y}_2^{\top}}{\|\mathbf{y}_2\|},\ldots,\frac{\mathbf{y}_N^{\top}}{\|\mathbf{y}_N\|}]^{\top},\quad \tilde{\mathbf{y}}=\frac{\hat{\mathbf{y}}}{\left\|\hat{\mathbf{y}}\right\|}, \tag{5}$$

then $d^{\ell}(\mathbf{t},\mathbf{y})=\left\|\tilde{\mathbf{t}}-\tilde{\mathbf{y}}\right\|^2$.

Note that the distance is the sum of reconstruction errors from different local regions

$$d^{\ell}(\mathbf{t},\mathbf{y})=\sum_{i=1}^{N}\frac{1}{N}\left\|\frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}-\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|}\right\|^2=\sum_{i=1}^{N}d_i^{\ell}(\mathbf{t},\mathbf{y}). \tag{6}$$

Each of the local reconstruction errors is bounded above:

$$d_i^{\ell}(\mathbf{t},\mathbf{y})=\frac{1}{N}\left\|\frac{\mathbf{t}_i}{\|\mathbf{t}_i\|}-\frac{\mathbf{y}_i}{\|\mathbf{y}_i\|}\right\|^2=\frac{2}{N}\left(1-\frac{\langle \mathbf{t}_i,\mathbf{y}_i\rangle}{\|\mathbf{t}_i\|\,\|\mathbf{y}_i\|}\right)\leq\frac{2}{N}. \tag{7}$$

This error bound restricts the negative influence of impulse noise to certain local parts and makes the distance metric not biased by the noisy parts. Therefore, we name this locally weighted distance with uniform weights as locally bounded distance metric. It can be seen from Figure 2 that the locally bounded distance are more robust than the Euclidean distance.

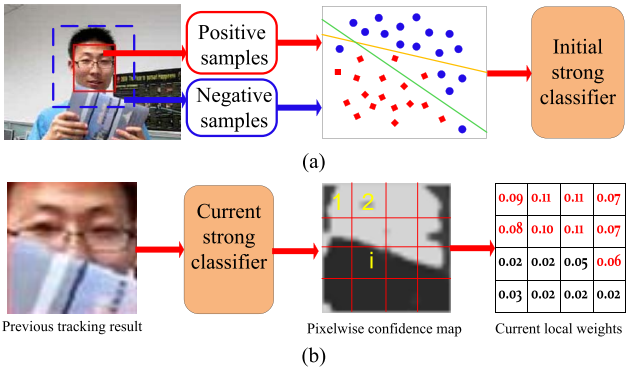


Fig. 3. (a) The RGB features of pixels are drawn as training samples. The initial strong classifier is obtained by combining weighted weak classifiers. (b) For the t-th frame, we test samples from the tracking result of $(t-1)$-th frame, and use the current strong classifier to compute the confidence map and the weights for local regions.

3) Learning Adaptive Local Weights: Given certain prior, the weights $\{W_i\}_{i=1}^N$ for local similarity terms $\{\rho(\mathbf{t}_i, \mathbf{y}_i)\}_{i=1}^N$ can be assigned adaptively to produce more accurate distances. In this work, the utilized prior is the temporal and spatial continuity of visual information, i.e., the similar target positions and impulse noise distributions between two consecutive frames. The local weights are learned mainly through two steps: to get the holistic confidence map of previous frame and to compute the weights based on the local statistic values of the map.

To obtain the holistic confidence map, we learn a boosted classifier overtime (with five weak classifiers) based on the Adaboost algorithm [10]. In the training process, the initial training samples are drawn from the first frame. Each sample is a 3D feature $\mathbf{f} \in \mathbb{R}^{3\times 1}$ that consists of the RGB values of the associated pixel. Feature vectors of all the pixels within the manually labeled target area (inside the red solid rectangle in Figure 3 (a)) are drawn as positive samples. While feature vectors of pixels in a surrounding region to the target (between the red solid rectangle and the blue dashed one in Figure 3 (a)) are labeled as negative samples. The size of negative sample set is made three times that of the positive one. Five weak classifiers are trained respectively in five iterations, each of which solves a weighted linear square regression to search a hyperplane separating positive samples from the negative ones. The strong classifier $H(\mathbf{f}) : \mathbf{f} \in \mathbb{R}^{3\times 1} \rightarrow \{-1, +1\}$ is then trained via the Adaboost algorithm.

For a frame at time t, the feature vectors composed of RGB values at every pixel of tracked object $\{\mathbf{f}_j\}_{j=1}^m$ in the $(t-1)$-th frame are classified by the learned boosted classifier $H_t(\mathbf{f})$, where m denotes the number of pixels and j represents the pixel index. Note that not all pixels within the rectangle target area belong to the target object especially when partial occlusion occurs. The classification margin $H_t(\mathbf{f}_j)$ is used to measure the confidence c_j of each pixel, and thus a holistic confidence map can be computed by,

$$c_j \propto \exp(-H_t(\mathbf{f}_j)), \tag{8}$$

where $H_t(\mathbf{f}_j)$ is obtained by setting negative margins to zero and rescaling the positive margins to the range $(0, 1]$.

After we have the confidence map of the $(t-1)$-th frame, we obtain the local similarity weight W_i at the t-th frame based on the statistical average of the confidence values within the i-th local part of the map:

$$C_i = \sum c_j, \quad \forall j \in \mathbf{I}_i, \quad i = 1, \ldots, N, \tag{9}$$

$$W_i = \frac{C_i}{\sum_{i=1}^{N} C_i}, \quad i = 1, \ldots, N, \tag{10}$$

where $\mathbf{I}_i$ denotes the set of pixel indexes in i-th local part.

These local weights $\{W_i\}_{i=1}^N$ indicate the probability of the i-th local part belonging to the target. The occluded parts have lower weights as shown in Figure 3 (b), and their negative influence are thus reduced. More experimental observations are demonstrated in Figure 6. We can see from this figure that the relatively stable parts of the target (like the upper body of a walking man) would be assigned larger weights when moderate shape deformation occurs. The parts with large weights facilitate object tracking since they are more credible in computing local similarity levels.

With this set of advanced local weights $\{W_i\}_{i=1}^N$, the locally weighted distance between $\mathbf{t}$ and $\mathbf{y}$ denoting as $d^\ell(\mathbf{t}, \mathbf{y})$ is rewritten as:

$$\begin{aligned} d^\ell(\mathbf{t}, \mathbf{y}) &= 2 - 2\sum_{i=1}^{N} W_i \times \rho(\mathbf{t}_i, \mathbf{y}_i) \\ &= \sum_{i=1}^{N} W_i \left\| \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} - \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|} \right\|^2 = \left\| \mathbf{w} \odot (\tilde{\mathbf{t}} - \tilde{\mathbf{y}}) \right\|^2, \end{aligned} \tag{11}$$

where $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d\}^\top$, $\mathbf{w}_j = W_i, \forall j \in \mathbf{I}_i$, and $\odot$ is the Hadamard product (element-wise product). With the adaptive local weights, errors from different local regions are bounded to more accurate upper bounds:

$$d_i^\ell(\mathbf{t}, \mathbf{y}) = W_i \left(2 - 2\frac{\langle \mathbf{t}_i, \mathbf{y}_i \rangle}{\|\mathbf{t}_i\| \, \|\mathbf{y}_i\|} \right) \leq 2W_i. \tag{12}$$

This local distance metric facilitates the proposed tracking algorithm to be robust during appearance change caused by partial occlusion and moderate shape deformation. That is, occluded or newly appearing local regions are assigned lower weights. In the meanwhile, the unchanged local regions have higher weights, thereby make them most important local regions when computing the distance.

To accommodate to the appearance changes of the target, we update the strong classifier with a standard way. After obtaining the current tracking result $\mathbf{r}_t$, we compute its distance from the current target template $\mathbf{t}_t$ in the locally weighted distance metric. If $\left\| \mathbf{w}_t \odot (\tilde{\mathbf{t}}_t - \tilde{\mathbf{r}}_t) \right\|^2 < \tau_1$ (a predefined constant), the strong classifier is updated by collecting new samples in the same way with initialization (Figure 3 (a)), substituting the old weak classifier with smallest weight for a newly trained one and reweighing all the weak classifiers. Otherwise, we do not update the classifier.

B. Tracking Based on the Inverse Sparse Representation

The particle filter framework [24] is employed in this work for its effective cooperation with the "sparse representation"-based formulation, as shown in many previous works [5], [12], [23], [34]–[36]. To keep the completeness of this work, we would first briefly review this framework. Then, we mainly focus on building an efficient and effective observation model by employing the inverse sparse representation formulation with the proposed distance metric.

1) The Particle Filter (PF) Framework: Particle filter is a Bayesian sequential importance sampling technique, which focuses on inferring the posterior distribution of state variables for a dynamic system. Usually, the PF framework uses a set of weighted particles to approximate the probability distribution of the state regardless of the underlying distribution. As a typical dynamic state inference problem, the tracking process can be achieved by using the PF framework [25].

In the PF framework, there exist two fundamental steps: prediction and update. Let $\mathbf{x}_t$ denote the state variable of the tracked object and $\mathbf{y}_t$ denote its corresponding observation in the t-th frame. Then the posterior probability can be recursively estimated by the following two rules:

$$p\left(\mathbf{x}_t|\mathbf{y}_{1:t-1}\right) = \int p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) p\left(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}\right) d\mathbf{x}_{t-1}, \quad (13)$$

$$p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right) = \frac{p\left(\mathbf{y}_t|\mathbf{x}_t\right) p\left(\mathbf{x}_t|\mathbf{y}_{1:t-1}\right)}{p\left(\mathbf{y}_t|\mathbf{y}_{1:t-1}\right)}, \quad (14)$$

where $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ stand for all available state vectors up to time t and $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$ denote their corresponding observations. $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a dynamic model that describes the state transition, and $p(\mathbf{y}_t|\mathbf{x}_t)$ is an observation model that estimates the likelihood of observing $\mathbf{y}_t$ at state $\mathbf{x}_t$. The posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is approximated by K weighted particles $\{\mathbf{x}_t^k, \beta_t^k\}_{k=1}^K$ drawn from an importance distribution $q(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1})$, which is chosen as $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in this work. The weight β_t^k is the observation likelihood of the i-th particle $\mathbf{x}_t^k$, which can be updated frame by frame as,

$$\beta_t^k = \beta_{t-1}^k \frac{p\left(\mathbf{y}_t^k|\mathbf{x}_t^k\right) p\left(\mathbf{x}_t^k|\mathbf{x}_{t-1}^k\right)}{q\left(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t}\right)}. \quad (15)$$

We use the affine transform to model $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$ of the tracked object. Let $\mathbf{x}_t = \{x_t, y_t, \theta_t, s_t, \delta_t, \phi_t\}$, where $x_t, y_t, \theta_t, s_t, \delta_t, \phi_t$ denote horizontal and vertical translations, rotation angle, scale, aspect ratio, and skew respectively. The state transition is formulated by random walk with a diagonalized Gaussian distribution. Finally, the optimal state $\mathbf{x}_t^*$ can be estimated by $\mathbf{x}_t^* = \int \mathbf{x}_t p\left(\mathbf{x}_t|\mathbf{y}_{1:t}\right) d\mathbf{x}_t \approx \sum_{k=1}^{K} \beta_t^k \mathbf{x}_t^k$ in the t-th frame.

2) Inverse Sparse Representation Formulation: Instead of using templates of a target object to represent each candidate region for tracking (as posed in existing algorithms [5], [12], [19], [22], [27], [35], [36]), we reverse their roles, using candidates to represent a target template. In addition, the proposed locally weighted distance metric is used to replace the Euclidean distance, facilitating more accurate reconstruction error measurement. The proposed formulation is intuitively shown in Figure 4.

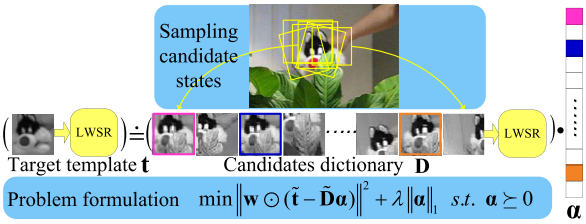


Fig. 4. The inverse sparse representation formulation. The dictionary is composed of candidate vectors and the target template is reconstructed by an ensemble of good candidates which correspond to the positive elements in $\boldsymbol{\alpha}$.

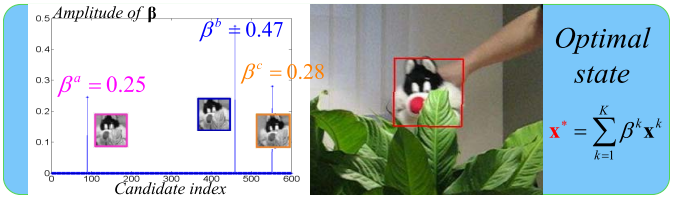


Fig. 5. The optimal state is obtained as a linear weighted combination of selected good candidates.

The initial target template $\tilde{\mathbf{t}}_1 \in \mathbb{R}^{d\times 1}$ is generated with the manually labeled truth in the first frame. The target image vector is locally decomposed following equation (5) and then weighted as described in Section II-A3. Then each local part of the target template is individually updated to maintain enough target information without degrading the template, which is elaborated later in the next subsection.

At the t-th frame, we draw K candidate states $\{\mathbf{x}_t^k\}_{k=1}^K$ with a particle filter in the neighborhood of previous target location. The observed image vectors $\{\mathbf{y}_t^k\}_{k=1}^K$ form the dictionary $\mathbf{D}_t = [\mathbf{y}_t^1, \mathbf{y}_t^2, \ldots, \mathbf{y}_t^K] \in \mathbb{R}^{d\times K}$ at time t.

Then we compute a non-negative sparse combination of dictionary atoms to reconstruct the template while minimizing reconstruction error in the locally weighted distance metric:

$$\boldsymbol{\alpha}_t^* = \arg\min_{\boldsymbol{\alpha}_t} \left\|\mathbf{w}_t \odot (\tilde{\mathbf{t}}_t - \tilde{\mathbf{D}}_t\boldsymbol{\alpha}_t)\right\|^2 + \lambda\|\boldsymbol{\alpha}_t\|_1 \quad \text{s.t. } \boldsymbol{\alpha}_t \succeq 0, \quad (16)$$

where λ is a penalty term, $\boldsymbol{\alpha}_t \succeq 0$ denotes the non-negativity constraint, and $\mathbf{w}_t$ is the weight vector obtained based on the $(t-1)$-th tracking result as described in Section II-A3.

In the theory of sparse representation, among all the dictionary subsets, the selected one most compactly expresses the input signal (the template in this case) and the atoms therein should be in the same class as the input signal [29]. That is, the candidates with positive elements in $\boldsymbol{\alpha}_t^*$ are the bases highly resemble to the target object. Here, we further assume that the magnitude of an element α_t^k measures the similarity between the target and the k-th candidate, as it is intuitive that the more a candidate resembles to the template the bigger contribution it should make to reconstructing the template. The experimental observations support this assumption as Figure 5 shows, and some examples can be found in section III-A.

Therefore, few candidates with larger magnitudes in $\boldsymbol{\alpha}_t^*$ should be more likely to belong to the target class and should be given larger weights when computing the optimum state,

whereas the other candidates corresponding to smaller or zero elements have low likelihoods belonging to the target class and should be assigned much smaller weights accordingly. Many candidate weights assigning schemes can be designed following the above rule, here, for efficiency we use a simple yet effective one, i.e., normalizing $\boldsymbol{\alpha}_t^*$ to obtain the observation likelihood of candidate states:

$$p\left(\mathbf{y}_t^k|\mathbf{x}_t^k\right) = \frac{\alpha_t^k}{\sum_{k=1}^{K}\alpha_t^k} = \frac{\alpha_t^k}{\|\boldsymbol{\alpha_t}\|_1}, \quad k = 1, 2, \ldots, K. \quad (17)$$

The observed sample corresponding to the obtained state is cropped out as the tracking result at the t-th frame (Figure 5).

Thus, the tracker merely requires to solve one ℓ_1 minimization problem per frame by using the proposed inverse sparse representation formulation with locally weighted metric. Thus, it much more time-saving than other sparsity based trackers like [5], [12], [19], [22], [27], and [36] where hundreds ℓ_1 minimization problems should be solved.

3) Online Update: In this work, a local update scheme is employed to accommodate target appearance change, in which each local part is individually updated by

$$\tilde{\mathbf{t}}_{t,i} = \mu\tilde{\mathbf{t}}_{t-1,i} + (1-\mu)\tilde{\mathbf{r}}_{t,i}, \quad \text{if } \left\|\tilde{\mathbf{t}}_{t-1,i} - \tilde{\mathbf{r}}_{t,i}\right\|^2 < \tau_2, \quad (18)$$

where the i-th local part of new target template $\tilde{\mathbf{t}}_{t,i}$ is the weighted combination of two local parts drawn from current tracking result $\tilde{\mathbf{r}}_{t,i}$, and the last stored target template $\tilde{\mathbf{t}}_{t-1,i}$ according to the weights assigned by the constant μ. The threshold τ_2 is an empirically defined parameter indicating the dissimilarity level. This scheme captures target appearance change even when heavy occlusion occurs. In such cases, the unoccluded local parts are still updated into the target template and the occluded ones are discarded.

III. Experiments

The proposed tracking algorithm is implemented in MATLAB on a PC with Intel i7-3770 CPU (3.4 GHz) and 32 GB memory. The parameters λ, μ, τ_1 and τ_2 are fixed to be 0.2, 0.95, 0.2 and 0.1 respectively based on empirical results. Each observed image patch is downsampled to 32×32 pixels. The size of local parts is set as 8×8 pixels from experimental results. All RGB features of the tracked region are employed when calculating the local weights.

In this section, we report our experimental observations from two aspects: self validation and comparison with the state-of-the-art trackers. We evaluate these algorithms on fifteen challenging video clips (among which two sequences consist of grayscale images[1]) to demonstrate the effectiveness of the proposed tracking algorithm using only intensity values. The state-of-the-art algorithms include both tracking methods based on sparse representation and other popular ones. The sparsity-based tracking algorithms include the adaptive structural local sparse appearance (ASLSA) [12], accelerated proximal gradient L1 (L1APG) [5], multi-task tracking (MTT) [35], online discriminative object tracking with local sparse representation (ODLT) [27], sparsity-based collaborative model (SCM) [36] and robust tracking using local sparse model and k-selection (SPT) [18] methods. The other evaluated state-of-the-art methods are the fragment-based tracking (Frag) [1], incremental visual tracking (IVT) [25], multiple instance learning (MIL) [4], tracking learning detection (TLD) [13], visual tracking decomposition (VTD) [14] and struck tracking (ST) [11] methods. Both qualitative and quantitative results are presented in this section.

[1]The MTT tracker, based on RGB features, is not tested on the grayscale sequences for fair comparisons.

TABLE I
Self Comparison With Reference Algorithms in Terms of Overlap Ratios

	IST	LBIST	LWIST -MAP	LWIST -MMP	LWIST -MSR	LWIST
Boy	0.75	0.78	0.53	0.80	0.76	0.79
Car11	0.44	0.58	0.65	0.76	0.77	0.77
Caviar1	0.30	0.88	0.89	0.92	0.91	0.91
Caviar2	0.60	0.78	0.78	0.79	0.79	0.78
David1	0.40	0.84	0.84	0.82	0.84	0.83
David2	0.44	0.45	0.69	0.78	0.75	0.75
Deer	0.15	0.58	0.58	0.63	0.61	0.61
Face1	0.90	0.90	0.91	0.91	0.91	0.91
Face2	0.44	0.85	0.79	0.83	0.83	0.85
Face3	0.34	0.85	0.83	0.85	0.83	0.82
Leno	0.86	0.72	0.72	0.90	0.85	0.86
Girl	0.58	0.63	0.40	0.63	0.64	0.63
Singer1	0.90	0.60	0.87	0.87	0.87	0.86
Sylv	0.21	0.78	0.66	0.84	0.85	0.79
Stone	0.44	0.66	0.59	0.65	0.75	0.66
Average	0.52	0.69	0.72	0.80	0.80	0.79
FPS	18	15	6	6	6	11

A. Self Validation

In this subsection, we present several reference algorithms for self comparison to provide a better understanding of the proposed tracker. The first one keeps the inverse sparse framework representation formulation, but use traditional holistic Euclidean distance metric, so we name it the inverse sparse tracker (IST). The second one use the locally bounded distance metric and keep other settings unchanged with the proposed tracker, therefore we name it the locally bounded inverse sparse tracker (LBIST). Others are the locally weighted inverse tracer with multiple templates with different pooling algorithms. These methods are tested over all fifteen image sequences, and the related results are reported in Table I.

From Table I, we can see that the LBIST method achieves a significant improvement from the IST method because the locally bounded distance metric is more robust to unexpected noise like partial occlusion and illumination changes as we have discussed in section II-A2. In addition, the LWIST method performs better thanks to the adaptively learned local weights, which further help the tracker focus on more credible parts of the target, as demonstrated in Figure 6 (a), (b), (c). They are three examples with moderate shape deformation caused by moving legs, partial occlusion and nonuniform illumination. The local weights are all accurately assigned: in (a) the upper parts of a human body (relative stable parts) are assigned larger weights; in (b), the unoccluded parts of the target are assigned larger weights; in (c), the parts under normal light condition are assigned larger weights.

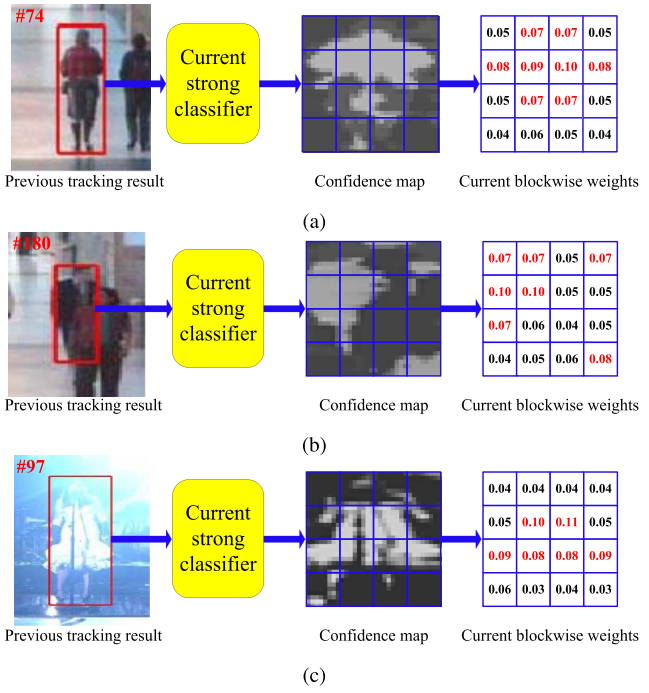


Fig. 6. Examples for learning adaptive weights, local parts with higher weights (in red) of each candidate are assumed to be more significant in the locally weighted distance metric. (a) Frame 75 in video *Caviar1* with shape deformation. (b) Frame 181 in video *Caviar2* with occlusions. (c) Frame 98 in video *Singer1* with nonuniform illuminations.

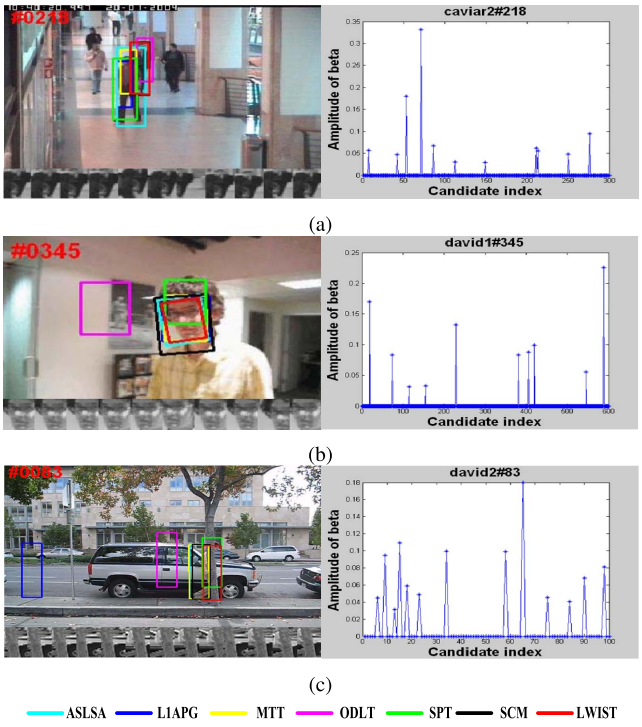


Fig. 7. Sampled results are shown in the left panel and the selected candidate appearances are presented at the bottom of the left panel. The weights of corresponding candidate states are shown in the right panel where the magnitude indicates the similarity of each candidate. (a) Frame 218 in video *Caviar2*. (b) Frame 345 in video *David1*. (c) Frame 83 in sequence *David2*.

Another reason why the LBIST and LWIST methods perform well is that with the accurate distance metric the inverse sparse formulation is more efficient for selecting good candidates. In Figure 7, we demonstrate some experimental observations regarding the formulation: some sampled tracking results (compared with other sparsity based trackers) as well as the cropped images of the selected candidate states with their weights, and the weights indeed increase with the similarity between candidate appearances and the target as we assume. In most cases, the usage of the proposed locally weighted distance metric facilitates selecting candidates similar to the target object, thereby ensuring the tracking accuracy of our method. Although a few bad candidates are selected, their weights are smaller than good ones as shown in the right panel.

In addition, a question might be raised that whether using multiple templates can improve our tracker. To address this issue, we design a self analysis experiment by comparing the proposed locally weighted inverse sparse tracker (LWIST) with its complex version that contains ten target templates $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{10}\}$. Each of the templates is reconstructed by candidates as described in equation (16), so that we have ten sparse coefficients $\{\alpha_1, \alpha_2, \ldots, \alpha_{10}\}$.[1] Then, three methods with multiple templates are presented based on the manner of using ten obtained coefficients, which include: (1) LWIST-MAP, using an average pooling scheme to combine coefficients; (2) LWIST-MMP, using a max pooling scheme to combine coefficients; and (3) LWIST-MSR, choosing the a coefficient vector with the smallest reconstruction error. The comparison results are represented in Table I, from which we can see that the LWIST trackers with multiple templates cannot improve the accuracy of the tracker significantly. Especially, the LWIST-MAP method performs worse that the LWIST one, since multiple templates accumulate more errors in the candidate weights by using the average pooling manner. Another disadvantage of LWIST trackers with multiple templates is that it is less efficient since reconstructing multiple templates requires more computational cost.

[1]To exclude the possibility that the templates choosing and updating might degrade the tracker, we use the same methods with [5] and [22] to initialize and update the templates, for those methods have been proven to be effective in the two previous work.

B. Quantitative Comparison

We assess the performance of trackers in terms of their center location errors, overlap ratios [9] and speed[2] in terms of frames per second (fps), which are the most widely used evaluation criteria. The numerical results are reported in Tables II-III. Overall, the sparsity-based trackers perform better in comparisons with other state-of-the-art methods. In particular, the proposed tracking method achieves favorable performance in terms of both accuracy and speed.

C. Qualitative Comparison

We present qualitative comparison with sparsity-based trackers in Figure 8, since they generally perform better and are more related to this work.

[2]The parameters in [5] are adjusted to obtain the best performance so the reported speed is different from that in [5].

TABLE II

AVERAGE CENTER LOCATION ERRORS (IN PIXEL). THE TOP THREE RESULTS ARE SHOWN IN RED, BLUE AND GREEN FONTS

	Frag	IVT	MIL	TLD	VTD	ST	ASLSA	L1APG	MTT	ODLT	SCM	SPT	LWIST
Boy	33.9	91.8	6.7	**4.5**	8.1	48.4	81.9	7.0	11.3	5.1	37.6	**2.2**	**3.5**
Car11	63.9	2.1	43.5	25.1	27.1	**1.3**	2.0	**1.7**	**1.8**	6.2	**1.8**	4.1	**1.7**
Caviar1	5.7	45.1	48.5	5.6	3.9	2.5	**1.4**	50.1	21.0	16.7	**0.88**	1.8	**1.1**
Caviar2	5.6	8.6	70.3	8.5	**4.7**	61.9	62.3	63.1	65.4	7.5	**2.5**	45.6	**1.1**
David1	148.7	**3.1**	34.3	13.4	49.4	35.5	3.5	10.8	13.4	34.3	**3.4**	6.3	**2.2**
David2	90.5	**52.3**	**38.4**	173.0	61.9	54.6	87.5	233.4	65.6	113.8	64.1	101.1	**5.4**
Deer	92.1	127.5	66.5	25.7	12.0	230.5	**8.0**	38.4	**9.2**	95.3	36.8	69.8	**10.8**
Face1	**5.6**	9.2	32.3	17.6	11.1	16.2	10.8	6.8	14.1	**4.3**	**3.2**	5.3	**4.3**
Face2	31.6	26.2	26.4	11.1	6.1	5.2	**2.9**	**3.4**	5.6	27.7	**2.9**	3.4	**3.2**
Face3	15.5	10.2	14.1	18.6	10.4	**3.1**	**3.1**	6.3	-	6.9	**4.8**	17.1	**4.5**
Girl	18.0	48.5	32.2	23.2	21.4	64.1	12.5	**11.1**	23.9	12.0	**9.7**	12.4	**10.8**
Leno	18.2	**6.3**	28.1	24.0	9.6	37.1	44.5	**5.9**	17.2	9.2	7.0	12.8	**5.7**
Singer1	22.1	8.5	15.2	32.7	4.1	21.9	5.3	**3.4**	41.2	4.5	**3.7**	14.5	**3.3**
Stone	65.9	**2.2**	32.3	8.0	31.4	3.3	**1.8**	6.4	-	42.2	2.4	68.7	**2.0**
Sylv	98.9	70.8	31.3	16.8	49.2	19.0	**9.1**	112.4	14.6	11.1	**9.0**	14.4	**5.3**
Average	47.7	34.2	34.7	27.2	**20.7**	40.3	22.4	37.3	23.4	26.5	**12.7**	28.1	**4.3**

TABLE III

AVERAGE OVERLAP RATES AND SPEEDS (IN fps). THE TOP THREE RESULTS ARE SHOWN IN RED, BLUE AND GREEN FONTS

	Frag	IVT	MIL	TLD	VTD	ST	ASLSA	L1APG	MTT	ODLT	SCM	SPT	LWIST
Boy	0.46	0.25	0.65	0.67	0.64	0.28	0.37	**0.74**	0.55	0.65	0.40	**0.82**	**0.79**
Car11	0.99	**0.81**	0.17	0.38	0.43	0.85	**0.81**	**0.83**	0.58	0.44	**0.79**	0.49	0.77
Caviar1	0.68	0.28	0.25	0.70	0.83	0.72	**0.90**	0.28	0.45	0.50	**0.91**	**0.85**	**0.91**
Caviar2	0.56	0.45	0.26	**0.67**	**0.67**	0.30	0.35	0.32	0.33	0.48	**0.81**	0.28	**0.78**
David1	0.09	0.69	0.23	0.5	0.23	0.43	**0.77**	0.63	0.53	0.40	**0.75**	0.72	**0.83**
David2	0.39	**0.52**	0.41	0.16	0.42	**0.53**	0.45	0.05	0.42	0.26	0.46	0.36	**0.75**
Deer	0.08	0.22	0.21	0.41	0.58	0.04	**0.62**	0.45	**0.60**	0.04	0.46	0.35	**0.61**
Face1	**0.90**	0.85	0.59	0.65	0.77	0.80	0.83	0.87	0.79	**0.91**	**0.93**	**0.90**	**0.91**
Face2	0.53	0.63	0.42	0.59	0.74	0.69	**0.87**	**0.86**	0.79	0.37	**0.85**	**0.85**	**0.85**
Face3	0.60	0.59	0.61	0.49	0.59	**0.87**	**0.81**	0.70	-	0.74	**0.82**	0.56	**0.82**
Girl	**0.69**	0.43	0.52	0.58	0.51	0.25	**0.72**	0.65	0.63	**0.68**	**0.69**	0.65	0.63
Leno	0.72	**0.85**	0.61	0.61	0.74	0.60	0.61	**0.86**	0.70	0.79	**0.84**	0.76	**0.86**
Singer1	0.34	0.66	0.34	0.41	0.79	0.35	0.78	**0.83**	0.32	0.73	**0.85**	0.52	**0.86**
Stone	0.15	**0.66**	0.32	0.41	0.42	0.54	0.56	**0.59**	-	0.07	**0.62**	0.09	**0.66**
Sylv	0.05	0.5	0.54	0.59	0.44	0.55	**0.77**	0.28	0.63	0.70	0.65	**0.73**	**0.79**
Average	0.48	0.56	0.41	0.52	0.59	0.52	**0.68**	0.60	0.56	0.52	**0.72**	0.60	**0.79**
Fps	4	**32**	**32**	**18**	4	15	9	4	1	0.05	0.5	2	**11**

1) David1 and Sylv: The *David1* sequence is challenging due to many factors: illumination change, shape deformation and scale change. The *Sylv* sequence includes not only similar challenging factors with *David1* but also in-plane rotation and occlusion when the target moves through the plant. Except the proposed algorithm, the other sparsity-based trackers all drift away from the target locations (frame 353 and frame 605 in *David1*; frame 364 in *Sylv*). The proposed algorithm accurately tracked the targets throughout these two sequences which can be explained with the usage of locally weighted distance metrics and sparse representation. These two schemes facilitate the tracker to weigh more on the stable blocks for selecting candidates when the targets undergo shape deformation. In addition, the local update manner enables the target template to better account for appearance change.

2) Deer and Boy: In the *Deer* sequence, there are considerable motion blurs caused by fast movements of the target object. Both the MTT and proposed LWIST algorithms are able to track the target while the others drift to different areas (frame 43 and frame 52 in *Deer*). The good performance of the LWIST algorithm can be attributed to that the similarity among multiple good candidates is considered in determining the optimal one, and thus the results are more robust than those based on a single candidate. Such results are more apparent when the target motion is abrupt as the possibility of drifting increases with the number of bad candidates. The *Boy* sequence is challenging due to the dancing movements. The proposed tracker performs well (frame 585) in this video clip despite pose change and fast motion.

3) Girl and Leno: In the *Girl* and *Leno* sequences, the targets move frequently with in-plane and out-of-plane rotations. The adopted affine motion model makes the proposed LWIST tracker accurately locate the location of the tracked object when in-plane rotation occurs (frame 312 in *Girl* and frame 464 in *Leno*). The ASLSA method fails to track the targets (frame 312 in *Girl* and frame 484 in *Leno*), since it is sensitive to the cluttered scenes where background pixels are included in the target area when the targets undergo

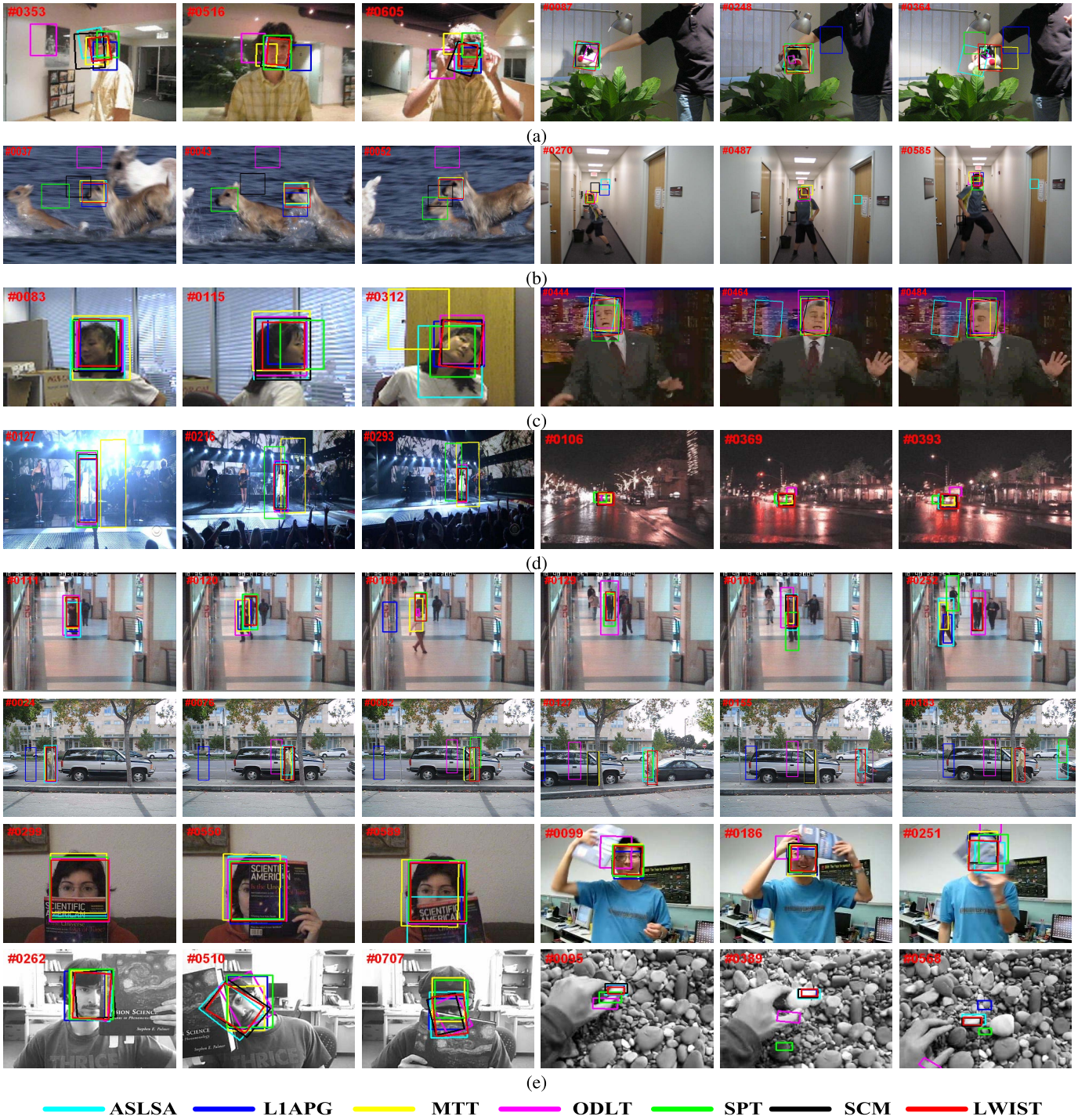


Fig. 8. Sampled tracking results of sparsity-based trackers. (a) *David* and *Sylv* sequences with dramatic deformation and illumination change. (b) *Deer* and *Boy* sequences with abrupt motion. (c) *Girl* and *Leno* sequences with in-plane and out-of-plane rotation. (d) *Singer1* and *Car11* sequences with severe illumination change. (e) *Caviar1*, *Caviar2*, *David2*, *Face1*, *Face2*, *Face3*, *Stone* sequences with varieties of occlusion and other challenging factors.

out-of-plane rotation. The proposed LWIST method is more robust to out-of-plane rotation as the blockwise errors are bounded by the proposed locally weighted sparse representation.

4) Singer1 and Car11: In the *Singer1* sequence, the illumination changes drastically due to stage lights and the target scale also changes rapidly. The MTT and SPT methods are less effective in handling dramatic illumination change and fail to locate the singer (frame 127 and frame 216). The proposed LWIST algorithm tracks the target stably, which can be attributed to that the locally weighted distance metric bounds blockwise reconstruction errors caused by nonuniform strong lights. In the *Car11* sequence, the tracked car moves in a night scene with low contrast. The OLDT method drifts away after the car makes a lane change (frame 369) because it mistakenly labels a background area as the tracking result.

5) Caviar1, Caviar2 and Stone: In the *Caviar1* and *Caviar2* sequences, the targets are often occluded by other

similar objects. Most trackers cannot handle occlusion well and drift away from the true targets (frame 111 in *Caviar1* and frame 195 in *Caviar2*). In the *Stone* sequence, the target object is surrounded by other pebbles with similar appearance. The ODLT and SPT methods mistakenly locate the hand as the target (frame 389) and the L1APG tracker drifts to other stones (frame 568). The SCM method and the proposed LWIST algorithm perform more robustly in these three sequences which can be attributed to the discriminative strength of classifiers to distinguish the foreground target from other similar objects in the background. The negative effects of partial occlusion are reasonably restrained.

6) Face1, Face2, Face3 and David2: The targets in the *Face1*, *Face2* and *Face3* sequences are heavily occluded (frame 550 in *Face1*, frame 251 in *Face2* and frame 707 in *Face3)*. Most sparsity-based algorithms track well in these sequences and the proposed tracker yields the most robust performance. The *David2* sequence is challenging as the target undergoes heavy occlusion and large pose variation. The L1APG method loses track of the target object right at the beginning when the person walks on the sidewalk, resulting in large appearance change when rectangular templates are used. It is difficult to correctly update object appearance with significant background noise for methods with the holistic representation (i.e., holistic rectangular templates). Although the ODLT, SCM and MTT methods are able to handle partial occlusion, these methods mistakenly update the models when the person walks behind the tree (frame 82) and consequently fail to track the target in the remaining frames. The ASLSA and SPT methods fail when the target turns around to walk back (frame 127 and frame 183), during which large appearance change makes it difficult for these update schemes to accommodate. The proposed LWIST tracker successfully locates the target throughout the sequence which can be attributed to the usage of the locally weighted distance metric for better handling occlusion and pose variation. In addition, the update scheme facilitates more accurate appearance update (i.e., updating reliable local parts when the target is occluded or deformed).

D. Effects of Key Parameters

In this subsection, we investigate the effects of some key parameters by using all video sequences, and report the average scores (i.e., overlap rates) and speeds (i.e., fps). We note that the average speeds with varied parameters are rescaled by $fps \leftarrow fps/11$, where 11 indicates the speed of the final LWIST method (reported in Table III).

1) Block Number: For the proposed LWIST method, the number of parts (or blocks) is a very important parameter. If the number of parts is too small, the LWIST algorithm is not able to achieve robust performance to deal with impulse noises (such as partial occlusion and local illumination variation). For another thing, if the number of parts is too large, each local part cannot capture enough contextual information, which makes the tracker be not stable. Figure 9 illustrate the tracking performance with different block numbers. Empirical results demonstrate that the proposed algorithm achieves best

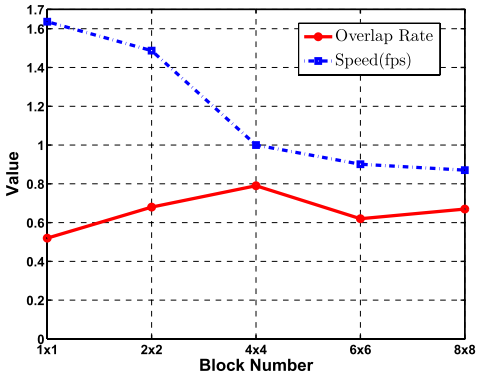


Fig. 9. The effects of different block numbers.

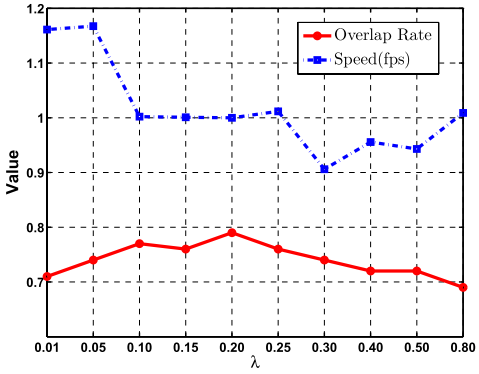


Fig. 10. The effects of varied λ values.

performance with 4×4 blocks to balance the accuracy and speed.

2) The Regularization Parameter λ: In the proposed tracker, the regularization parameter λ is also a very critical parameter, which controls the sparsity level of our inverse sparse representation formulation. If the value of λ is too small, many candidate will be maintained. It may introduce lots of unexpected noises and lead to an unstable solution. If λ is too large, the sparsity will be over-emphasized. It may make solution be not suitable for maintaining the variety of particles. In Figure 10, the accuracies and speeds with different λ values are reported, from which we can see that our tracker is able to achieve a satisfying result when $\lambda = 0.2$.

3) Parameters τ_1, τ_2, μ: The parameters τ_1, τ_2 and μ focus on controlling the update scheme in our tracker. Figure 11 demonstrates the effects of these parameters, from which we have two obvious observations: (1) The parameter τ_1 controls the update frequency of the Adaboost classifier. It can be seen from Figure 11 (a) that the tracking speeds drop out significantly with the increase of τ_1 without improving the tracking accuracies. (2) From Figure 11 (b), we can see that our tracker performs worse if $\tau_2 > 0.1$. The underlying reason is that the template of the tracked object will be updated by some noise observed samples if τ_2 is too large.

E. Limitations and Failure Cases

From Section IV.B-C, we can see that the proposed tracking algorithm is able to effectively and efficiently deal with many

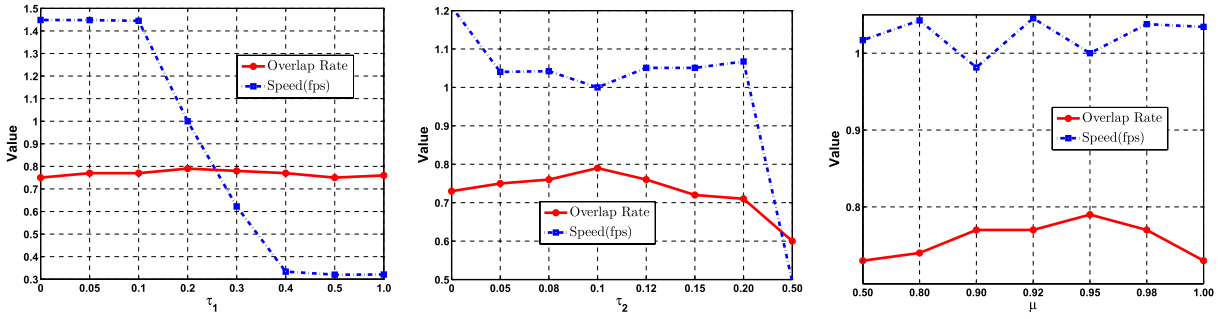
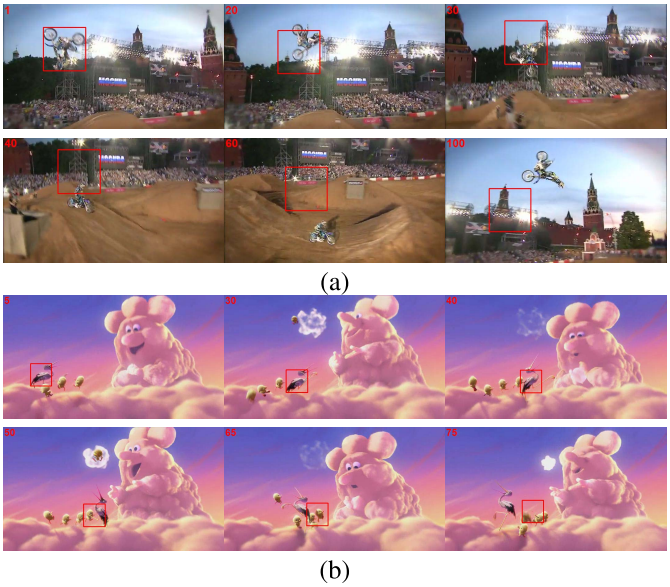


Fig. 11. The effects of parameters τ_1, τ_2 and μ.



(a)

(b)

Fig. 12. An illustration of some failure cases. (a) *MotorRolling*. (b) *Bird*.

challenging factors including partial occlusion, illumination variation, slight pose change, motion blur, background clutter and so on. This can be mainly attributed to the proposed locally distance metric and the related weight learning scheme. However, our tracker cannot well handle challenging factors caused by large non-rigid transformation and heavy pose change (as illustrated in Figure 12), which is the limitation of the proposed method. In addition, the proposed tracker also performs not good when the tracked object is lost or out of the view, since it is not equipped with a re-initialization mechanism.

IV. Conclusions

In this paper, we present an inverse sparse tracker with a locally weighted distance metric. The error contribution from each local part is restricted to a weighted upper bound based on the proposed locally weighted distance. This scheme makes the proposed tracker robust to partial occlusion or nonuniform illumination change as well as moderate shape deformation. In addition, we employ the inverse sparse formulation which achieves tracking results by better exploiting the compactness and uniqueness properties of sparse representation coefficients. The formulation facilitates solving only one ℓ_1 minimization problem per frame, thereby facilitating an efficient tracker and the proposed locally weighted metric improves its robustness. Numerous experimental observations are presented, from which the key components of our tracker can be better comprehended. Quantitative and qualitative experimental results on challenging sequences demonstrate the effectiveness and efficiency of the proposed tracking algorithm.

References

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

[2] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Proc. Eur. Conf. Comput. Vis.*, May 2002, pp. 113–130.

[3] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using CamShift algorithm and multiple quantized feature spaces," in *Proc. Pan-Sydney Area Workshop Vis. Inf. Process.*, 2004, pp. 3–7.

[4] 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.

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

[6] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *Proc. Int. Conf. Image Process.*, vol. 3. Sep. 2000, pp. 70–73.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[8] F. Yang, H. Lu, and M. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

[9] 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, Jun. 2010.

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. EuroCOLT*, vol. 904. 1995, pp. 23–37.

[11] 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.

[12] 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.

[13] Z. Kalal, J. Matas, and K. Mikolajczyk, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

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

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

[16] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1305–1312.

[17] 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.

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

[19] 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. Eur. Conf. Comput. Vis.*, 2010, pp. 624–637.

[20] H. Liu and F. Sun, "Visual tracking using sparsity induced similarity," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1702–1705.

[21] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[22] 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.

[23] X. Mei, H. Ling, Y. Wu, E. P. Blasch, and L. Bai, "Efficient minimum error bounded particle resampling ℓ_1 tracker with occlusion detection," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2661–2675, Jul. 2013.

[24] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2002, pp. 353–360.

[25] 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, Aug. 2008.

[26] H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 75–89, Jan. 2002.

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

[28] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[29] 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.

[30] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[31] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[32] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1531–1536, Nov. 2004.

[33] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 864–877.

[34] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 470–484.

[35] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.

[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] 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.

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

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

Dong Wang received the B.E. degree in electronic information engineering and the Ph.D. degree in signal and information processing from the Dalian University of Technology (DUT), Dalian, China, in 2008 and 2013, respectively. He is currently a Faculty Member with the School of Information and Communication Engineering, DUT. His current research interests include face recognition, interactive image segmentation, and object tracking.

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), Dalian, China, in 1998 and 2008, respectively. He has been with the School of Information and Communication Engineering, DUT, as a Faculty Member, since 1998, and a Professor since 2012. His current research interests include computer vision, pattern recognition, visual tracking, and segmentation. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS.

Ziyang Xiao received the B.E. degree in electronics engineering and the master's degree from the School of Information and Communication Engineering, Dalian University of Technology, China, in 2011 and 2014, respectively. Her research interest is in object tracking.

Ming-Hsuan Yang (SM'06) received the Ph.D. degree in computer science from the University of Illinois at Urbana—Champaign, in 2000. He is currently an Associate Professor of Electrical Engineering and Computer Science with the University of California at Merced (UC Merced). He is a Senior Member of the Association for Computing Machinery. He was a recipient of the NSF CAREER Award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He served as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2007 to 2011, and is an Associate Editor of the *International Journal of Computer Vision, Image and Vision Computing* and the *Journal of Artificial Intelligence Research*.