# Robust Object Tracking via Key Patch Sparse Representation

Zhenyu He, *Senior Member, IEEE*, Shuangyan Yi, Yiu-Ming Cheung, *Senior Member, IEEE*, Xinge You, *Senior Member, IEEE*, and Yuan Yan Tang, *Fellow, IEEE*

***Abstract*—Many conventional computer vision object tracking methods are sensitive to partial occlusion and background clutter. This is because the partial occlusion or little background information may exist in the bounding box, which tends to cause the drift. To this end, in this paper, we propose a robust tracker based on key patch sparse representation (KPSR) to reduce the disturbance of partial occlusion or unavoidable background information. Specifically, KPSR first uses patch sparse representations to get the patch score of each patch. Second, KPSR proposes a selection criterion of key patch to judge the patches within the bounding box and select the key patch according to its location and occlusion case. Third, KPSR designs the corresponding contribution factor for the sampled patches to emphasize the contribution of the selected key patches. Comparing the KPSR with eight other contemporary tracking methods on 13 benchmark video data sets, the experimental results show that the KPSR tracker outperforms classical or state-of-the-art tracking methods in the presence of partial occlusion, background clutter, and illumination change.**

***Index Terms*—Occlusion prediction scheme, particle filter, patch sparse representation, template update, visual object tracking.**

Manuscript received July 9, 2015; revised October 25, 2015; accepted December 23, 2015. Date of publication March 11, 2016; date of current version January 13, 2017. This work was supported in part by the Shenzhen Research Council under Grant JSGG20150331152017052 and Grant JCYJ20140819154343378, in part by the Faculty Research Grant of Hong Kong Baptist University (HKBU) under Project FRG2/12-13/082, Project FRG1/14-15/041, and Project FRG2/14-15/075, in part by the Knowledge Transfer Office of HKBU under Grant MPCF-005-2014/2015, in part by the National Natural Science Foundation of China under Grant 61272366 and Grant 61272203, in part by the National Science and Technology Research and Development Program under Grant 2015BAK36B00, and in part by the Hubei Province Science and Technology Support Program under Grant 2013BAA120. This paper was recommended by Associate Editor M. Shin.

Z. He is with the School of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China (e-mail: zyhe@hitsz.edu.cn).

S. Yi is with the School of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China, and also with the Institute of Research and Continuing Education, Hong Kong Baptist University, Hong Kong.

Y.-M. Cheung is with the Department of Computer Science and the Institute of Research and Continuing Education, Hong Kong Baptist University (HKBU), Hong Kong, and also with the United International College, Beijing Normal University—HKBU, Zhuhai 519000, China (e-mail: ymc@comp.hkbu.edu.hk).

X. You is with the Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Research Institute of Huazhong University of Science and Technology in Shenzhen, Shenzhen 518057, China (e-mail: youxg@hust.edu.cn).

Y. Y. Tang is with the Faculty of Science and Technology, University of Macau, Macau 999078, China (e-mail: yytang@umac.mo).

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/TCYB.2016.2514714

## I. Introduction

OBJECT tracking, which plays an indispensable role in motion analysis, activity recognition, video surveillance, and traffic monitoring, continually attracts attention in the computer vision community. Though numerous tracking methods have been proposed for object tracking in the past decades, it still remains a challenging problem because of many environmental factors in video data sets, such as illumination variation, background clutter, and occlusions.

Generally, object tracking methods can be classified into discriminative and generative methods. The discriminative methods [1]–[8] aim to discriminate the target from the background by training a classifier according to the information from both the target and the background. The generative methods [9]–[18] aim to search for regions, which are extremely similar to the target, based on templates or subspaces.

In discriminative methods, support vector tracking [7] is proposed by integrating support vector machines (SVMs) [6] into an optic-flow-based tracker and maximizing the classification score. Multiple instance learning (MIL) [2] is trained with instances, which are included in the bags. The MIL problem can be cast as a maximum margin problem and solved by SVM. P–N learning (PN) [3], which is guided by positive and negative constraints on the unlabeled data, is proposed to exploit the underlying structure of positive and negative samples to learn effective classifiers for object tracking. However, all of these existing methods based on classification rely on a heuristic intermediate step for producing labeled binary samples, which is often a source of error during tracking. Therefore, a new adaptive tracking-by-detection framework based on structured output prediction [8] is proposed, which is able to avoid the need for an intermediate classification step and incorporate image features and kernels.

In generative methods, it is necessary and difficult to solve partial occlusion. Adam *et al.* [11] adopted a patch-based tracking method to allow every patch vote on the possible positions and scales of the target and then locate it by combining the vote maps of the multiple patches. The patch-based tracking methods can solve the partial occlusion to some extent. The sparsity-based tracking methods [16], [19]–[21], inspired by face recognition [22], [23] play an important role in object tracking. Mei and Ling [16] first formulated the tracking problem as a sparse approximation problem. And then Wu *et al.* [20] proposed a data fusion approach via

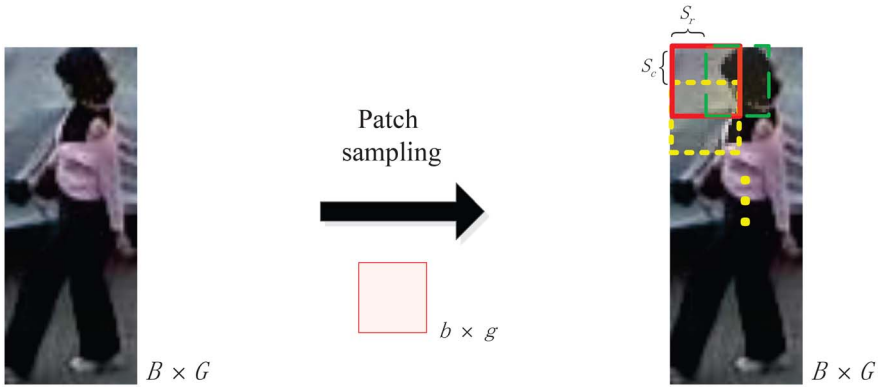



Fig. 1. Patch sampling. Given the target candidate with $B \times G$ pixels (left), the sampled patch with $b \times g$ pixels (middle), and step length $S_r$ and $S_c$ (right), the patches with different colors and linetypes are formed in the right subfigure.

sparse representation, where a flexible framework is provided and the information from different data sources can be easily integrated. The incremental visual tracking (IVT) [15] is robust to illumination and pose variation but sensitive to partial occlusion and background clutter. Naturally, inspired by [11], [15], and [16], visual tracking via adaptive structural local sparse appearance (ASLSA) model [18] is proposed that uses the patch sparse representations to deal with the partial occlusion to some extent.

Although the aforementioned methods have achieved a prominent performance in many cases, their performances can also be further improved since these methods do not consider the local information or do not consider the difference among the patches sampled from a target candidate and treats them equivalently. To this end, we propose a robust tracker based on key patch sparse representation (KPSR). The proposed method is based on patches and treats them differently. Its main contributions are twofold. First, we propose a selection criteria based on the key patch according to occlusion prediction and patch location. Second, we propose a contribution factor design for key patch and nonkey patch regions and emphasize the contribution of key patch for robust tracking.

The remainder of this paper is organized as follows. In Section II, we first introduce patch sparse representation to get the score of each patch, and then propose the KPSR for tracking in Section III. In Section IV, we propose the robust tracker based on KPSR. In Section V, we make quantitative and qualitative evaluations, and compare the KPSR with eight tracking methods, including the classical and state-of-the-art tracking methods. Finally, the conclusion is drawn in Section VI.

## II. Preliminary

In this section, we first give the general formulation of patch sampling and then generalize the process of patch sparse representation.

### A. Patch Sampling

Given the target candidate with $B \times G$ pixels, the patch with $b \times g$ pixels, the column step length $S_c$ (an integer), $0 < S_c \leq b$, and the row step length $S_r$ (an integer), $0 < S_r \leq g$, the desired patches are sampled sequentially and used to represent the complete structure of the target candidate (Fig. 1). In detail, if $\mathrm{mod}(B-b, S_c) = 0$ and $\mathrm{mod}(G-g, S_r) = 0$, we can get $(((G-g)/(S_r))+1)$ patches in row orientation and $(((B-b)/(S_c))+1)$ patches in column orientation; respectively. Let $r = ((G-g)/(S_r))+1$ and $c = ((B-b)/(S_c))+1$, then the number of total patches of a target candidate is $N = rc$. Now, the serial number of each patch $k$, $k \in \{1, 2, \ldots, N\}$ can be denoted as $k = (j-1)r + i$, where $i$, $i = 1, 2, \ldots, r$ is row index and $j$, $j = 1, 2, \ldots, c$ is column index. To sum up, given the target size, the sampled patch size, and the step length, key patch sampling will result in $N$ patches.

Note that patch sampling with different step length differ. When $S_c = b$ and $S_r = g$, it becomes the unoverlapped patch sampling, otherwise it becomes the overlapped patch sampling. Usually, the overlapped patch sampling is adopted because it is able to better capture local structure. In this paper, the bounding boxes of the target candidates are first resized to $32 \times 32$ pixels, i.e., $B = G = 32$. The patch size is $b \times g = 16 \times 16$ pixels, and the step length is $S_r = S_c = 8$, which will lead to an overlapped patch sampling strategy and finally get $N = 9$ overlapped patches.

Adam *et al.* [11] divided the target bounding box into several unoverlapped vertical and horizontal patches. On the contrary, our patch sampling strategy is similar to that of [11] and [18], which utilizes an overlapped patch sampling strategy, thus can keep more spatial structural information between the adjacent patches. Finally, we should notice that the number of patches depends more on the experience and the consideration of the balance between accuracy and speed.

### B. Patch Score via Sparse Representation

First, getting a dictionary is necessary. We manually label the target in the first frame and use $K$-dimensional tree [24] to track the target from the second frame to the $n$th frame. After, we use the tracked target up to $n$ frames to initialize a template set $\hat{T}_n = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_n\}$, where $\mathbf{T}_i$ with $B \times G$ pixels is obtained by the tracked target in the $i$th frame and $n$ is the template number in template set. Then, each template in $\hat{T}_n$ is divided into $N$ patches with a spatial layout (Fig. 1),

and these patches are assembled into a patch-dictionary $\hat{D} = \{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_{n\times N}\} \in \mathbb{R}^{d\times(n\times N)}$, where $d = b \times g = 256$ is the dimension of each patch after turning into vector.

Next, the target is tracked in the $n+1$th frame. $M$ target candidates are sampled surrounding the tracked target in the $n$th frame. For a target candidate, we divide it into $N$ patches and turn them into vectors, which are denoted as $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. With sparse representation, the candidate patch $\mathbf{y}_k \in \mathbb{R}^{d\times 1}$ can be represented by $\hat{D}$, and the corresponding coefficient vector $\mathbf{z}_k$ can be obtained by solving the following Lasso problem using the corresponding Lasso [25]–[27] method:

$$\begin{aligned} \min_{\mathbf{z}_k} \quad & \left\|\mathbf{y}_k - \hat{D}\mathbf{z}_k\right\|_2^2 + \lambda \parallel \mathbf{z}_k \parallel_1, \ k = 1, 2, \ldots, N \\ \text{s.t.} \quad & \mathbf{z}_k \succeq \mathbf{0} \end{aligned} \tag{1}$$

where the vector $\mathbf{z}_k \in \mathbb{R}^{(n\times N)\times 1}$ is the corresponding sparse coefficients of $\mathbf{y}_k$, and $\mathbf{z}_k \succeq \mathbf{0}$ means each element of $\mathbf{z}_k$ is nonnegative. According to the different templates, $\mathbf{z}_k$ is divided into $n$ group vectors, i.e., $\mathbf{z}_k^\top = [\mathbf{z}_k^{(1)\top}, \mathbf{z}_k^{(2)\top}, \ldots, \mathbf{z}_k^{(n)\top}]$. Here, $\mathbf{z}_k^{(i)} \in \mathbb{R}^{N\times 1}$ means the $i$th group vector of $\mathbf{z}_k$, $i = 1, 2, \ldots, n$. Then, we compress $\mathbf{z}_k^\top$ and get $\mathbf{v}_k \in \mathbb{R}^{N\times 1}$ as follows:

$$\mathbf{v}_k = \frac{1}{C}\sum_{i=1}^{n} \mathbf{z}_k^{(i)}, k = 1, 2, \ldots, N \tag{2}$$

where $C$ is a normalization constant. Note that a square matrix $\mathbf{V}$ is formed by $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$.

In the same way, all $M$ target candidates get the corresponding $M$ square matrices. In order to distinguish between different target candidates, $\mathbf{v}_{l_k} \in \mathbb{R}^{N\times 1}$ is denoted as the coefficient vector of the $k$th patch for the $l$th target candidate (patch $k$ in $l$), $l = 1, 2, \ldots, M$, and $v_{l_{kk}}$ (the $k$th element of vector $\mathbf{v}_{l_k}$) is selected as the patch score of patch $k$ in $l$. The target candidate $l$ with the maximum sum is chosen as the target in the $n+1$th frame as follows:

$$\bar{\mathbf{E}}_l = \max_l \left\{\sum_{k=1}^{N} v_{l_{kk}}\right\}. \tag{3}$$

## III. Key Patch Sparse Representation

Unfortunately, patch sparse representation in [18] does not consider the different contributions among these patches, and this may result in the drift when partial occlusion or some background information exist in the bounding box. In view of this, we propose KPSR to reduce the effect of occlusion or background clutter. Generally, the target lies in the center of the bounding box including target information and little background information. Therefore, it is reasonable that the middle patch (such as the green box in the right subfigure of Fig. 1) should account for a larger contribution and the peripheral patch (such as the red box in the right subfigure of Fig. 1) accounts for a smaller contribution. On the other hand, when occlusions exists, the corresponding patch should account for a smaller contribution. Therefore, our idea is to select the key patch according to the location and occlusion case of each patch, and then design the patch's contribution factor for the key patch and nonkey patch.

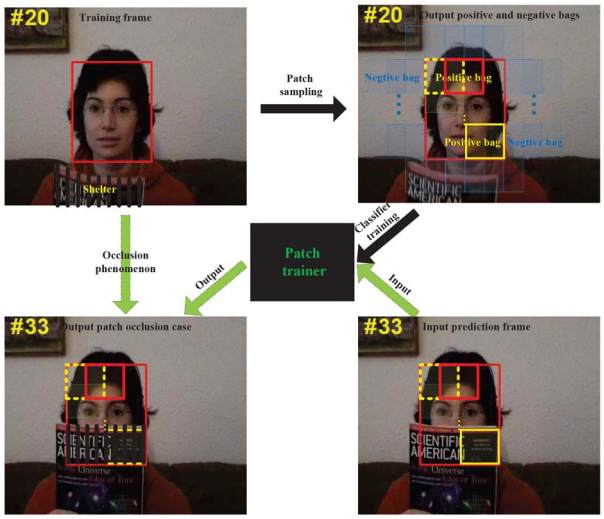



Fig. 2. Occlusion prediction scheme. We first use a set of black arrows to represent the process of classifier training and then use it to predict the occlusion case of each patch of the tracked target. Taking the 20th frame for example, we take the overlapped patch samples for the bounding box and assemble the sampled patches in each row as a bag. Similarly, we also take the overlapped patch samples for the background, which are labeled by the blue squares. In this way, the sampled patches in each row within the bounding box form a positive bag and the sampled patches in each row outside the bounding box form a negative bag. Then, we use SVM to train the patches and get the patch trainer, which can be used to predict the occlusion case of the target in the 33rd frame.

### A. Selection of Key Patch

Consider that the target usually lies in the middle of the bounding box, we select the middle patch as key patch. In the below, we discuss how to select the key patch when the tracked target suffers from occlusion.

*1) Occlusion Prediction Scheme:* We typically observe that occlusion phenomenon results from the background information. Here, background information is regarded as the image information except for the target information. That is to say, occlusion happens if some background information enters the bounding box. Fig. 2 gives an occlusion phenomenon, where we can observe that the shelter (i.e., a book) gradually enters the bounding box and covers one-third of the woman in the 33rd frame. Our goal is to predict the occlusion the shelter in the bounding box.

Inspired by MIL [2], we propose an occlusion prediction scheme. We define each patch as an instance and require the positive bag includes at least one positive instance and all instances of the negative bag are negative. In fact, the patch sampling in Section II should be regarded as inner patch sampling where the patches are sampled in the bounding box. Besides the inner patch sampling, we adopt the outer patch sampling that is to sample patches surrounding the bounding box. Taking Fig. 2 for example, we first make the inner patch samples and define the patches in a row as a positive bag. As the height of the bounding box nearly equals to the height of the target, inner patch sampling can ensure all bags sampled from the bounding box are positive. Then, we make outer patch samples and still define the patches in a row as a negative bag. After we get the positive and negative bags

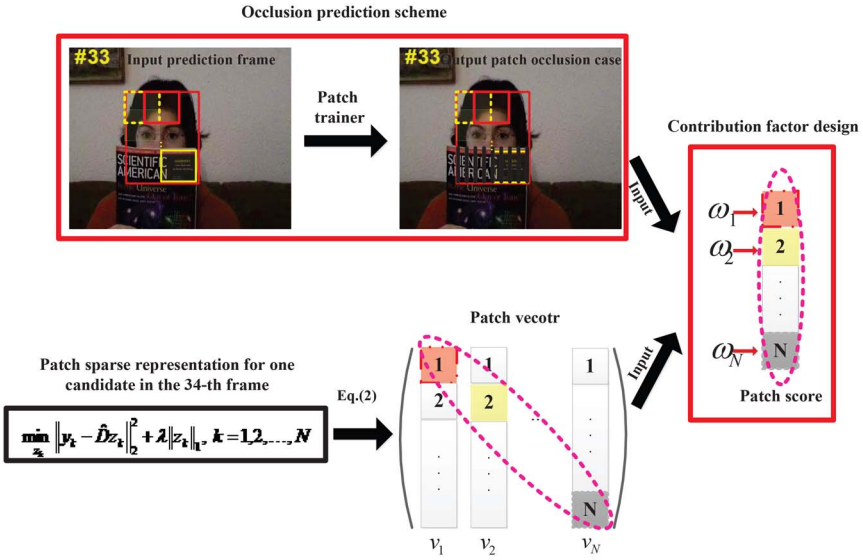



Fig. 3. Overview of KPSR. For a candidate, we first adopt the patch sparse representation to get the patch score, and then design the corresponding patch's contribution factor. The two big red rectangles represent our main contribution, which is the difference between ASLSA and KPSR.

which are used to label its including patches, MIL&SVM [6] is used to train the patches and a classifier (patch trainer) is obtained. With the classifier, we can predict the occlusion case of each patch, denoted by a binary indicator vector $\delta_k, k = 1, 2, \ldots, N$. If patch $k$ is not occluded, $\delta_k = 1$, otherwise $\delta_k = 0$. Therefore, when the current tracked target suffers from occlusions, we can find those occluded patches according to $\delta_k = 0, k = 1, 2, \ldots, N$.

It is worth noting that the initial classifier is formed by exploring the tracked target across the initial $n$ frames. As the appearance of the target changes, the classifier update is necessary. In this paper, the classifier is updated every $\theta$ frames when the target does not have a severe occlusion.

### B. Contribution Factor Design

After key patch is selected, we propose a contribution factor design that assigns different contribution factors for the sampled patches and emphasizes the contribution of key patch. In order to keep consistence with occlusion indicator $\delta_k$, we use $\omega_k, k = 1, 2, \ldots, N$ to represent the contribution of total $N$ patches. The contribution factor of patch $k$, $\omega_k$, is defined as follows:

$$\omega_k = 1 + \delta_k e^{-\beta\left(\left|i-\frac{1+r}{2}\right|+\left|j-\frac{1+c}{2}\right|\right)}, \quad \begin{matrix} i = 1, 2, \ldots, r \\ j = 1, 2, \ldots, c \end{matrix} \tag{4}$$

where $\delta_k$ means the occlusion indicator of patch $k$, $\beta$ is a constant, $r$ is the patch's number in a row, and $c$ is the patch's number in a column.

When the target does not suffer from occlusion (i.e., all total $N$ patches do not suffer from occlusion), we design the contribution factor as $\omega_k = 1 + e^{-\beta(|i-((1+r)/2)|+|j-((1+c)/2)|)}, k = 1, 2, \ldots, N$, where the term $e^{-\beta(|i-((1+r)/2)|+|j-((1+c)/2)|)}$ indicates that different patch locations have different contribution factors. In detail, the patch [i.e., $i = ((1 + r)/2)$ and $j = ((1 + c)/2)$] has a largest contribution factor 2. The peripheral patch (e.g., $i = 1$ and $j = 1$) have such a contribution factor that is smaller than 2 and larger than 1. When the target suffers from a partial occlusion, the occluded patch $k$ has $\delta_k = 0$ and hence its contribution factor is $\omega_k = 1$. When the target suffers from a complete occlusion, the contribution factor of each patch becomes 1. To sum up, the contribution factor of each patch considers not only its location but also its occlusion case. In this way, the contribution of the middle and unoccluded patch is emphasized. Fig. 3 gives the overview of KPSR.

## IV. Robust Tracker Based on Key Patch Sparse Representation

We first combine KPSR with particle filter to construct the objective function of the tracker. Second, we give a template update to adapt to the appearance change of the target. Finally, we discuss the robust tracker under three cases including no occlusion, partial occlusion, and complete occlusion.

### A. Objective Function

We use the status variable $\mathbf{E}^t$ to represent the location and shape of the target to be tracked in the $t$th frame and $\mathbf{A}^{1:t} = \{\mathbf{A}^1, \ldots, \mathbf{A}^t\}$ to represent the observation set of the target from the first frame to the $t$th frame. Target tracking is to estimate a posterior probability $p(\mathbf{E}^t|\mathbf{A}^{1:t})$ that can be written as follows:

$$p\left(\mathbf{E}^t\middle|\mathbf{A}^{1:t}\right) \propto p\left(\mathbf{A}^t\middle|\mathbf{E}^t\right)\int p\left(\mathbf{E}^t\middle|\mathbf{E}^{t-1}\right)p\left(\mathbf{E}^{t-1}\middle|\mathbf{A}^{1:t-1}\right)d\mathbf{E}^{t-1} \tag{5}$$

where $p(\mathbf{A}^t|\mathbf{E}^t)$ means the observation model in the $t$th frame and describes the similarity between a target candidate and the target templates. $p(\mathbf{E}^t|\mathbf{E}^{t-1})$ means the motion model in the successive frames and describes the temporal correlation of the target state. $\mathbf{E}^t = (x, y, \theta, s, \beta, \phi)$ is consisted of six parameters of the affine transformation, where $x, y, \theta, s, \beta$, and $\phi$ denote 2-D translations, rotation angle, scale, aspect ratio, and skew, respectively. In this paper, the motion model

**Algorithm 1** Template Selection

**Input**: Old template set $\hat{T}_{f-1}$ including $n$ templates, observation vector $\mathbf{p}$, eigenbasis vectors $\mathbf{U}$, occlusion ratio $r_{occ}$, threshold $\eta$, the current frame $f$ ($f > n$);

1: Take $M$ candidates surrounding the $f-1$-th frame, and use (Eq. (1)) to obtain patch score (see Section II-B);
2: Use Eq. (8) to ascertain the optimal candidate $\bar{\mathbf{E}}_l$ as the target in the $f$-th frame;
3: Update template to get $\hat{T}_f$;
- If $mod(f, 5) = 0$ and $r_{occ} \leq \eta$
- Generate a random number between 0 and 1 and decide the discarded template $\mathbf{T}_d$, then $\hat{T}_f^{lack} = \hat{T}_{f-1} - \mathbf{T}_d$;
- Solve Eq. (10) to obtain $\mathbf{q}$;
- Add $\mathbf{p} = \mathbf{Uq}$ into template set $\hat{T}_f^{lack}$ and output the new template set $\hat{T}_f = \hat{T}_f^{lack} + \mathbf{p}$;
- Else $\hat{T}_f = \hat{T}_{f-1}$;
- End

**Output**: New template set $\hat{T}_f$.

$p(\mathbf{E}^t|\mathbf{E}^{t-1})$ is modeled as $p(x_t|x_{t-1}) = N(x_t; x_{t-1}, \sigma)$, where $\sigma$ is a diagonal covariance matrix and the diagonal elements are the variances of the affine parameters. For each target candidate in the particle filter framework, estimating the posterior probability $p(\mathbf{E}^t|\mathbf{A}^{1:t})$ is converted into maximizing $p(\mathbf{A}^t|\mathbf{E}^t)$.

In our tracker based on KPSR, we replace $p(\mathbf{A}^t|\mathbf{E}^t)$ with $p(\mathbf{A}^t|\delta^t, \mathbf{E}^t)$. In detail, we use $\mathbf{E}_l^t$ to represent the state of the target candidate $l$ in the $t$th frame and $\mathbf{E}_{l_k}^t$ to represent the state of patch $k$ in $l$, $l = 1, 2, \ldots, M$. Then, for each state of target candidate $l$, we have $p(\mathbf{A}_l^t|\delta_l^t, \mathbf{E}_l^t) = \prod_{k=1}^N p(\mathbf{A}_l^t|\delta_{l_k}^t, \mathbf{E}_{l_k}^t)$, where $\delta_{l_k}^t$ is the occlusion prediction indicator of patch $k$ in $l$. Without loss of generality, we remove the frame index $t$ and have $p(\mathbf{A}_l|\delta_l, \mathbf{E}_l) = \prod_{k=1}^N p(\mathbf{A}_{l_k}|\delta_{l_k}, \mathbf{E}_{l_k})$. Consequently, our objective function $\bar{\mathbf{E}}_l$ can be written as

$$\bar{\mathbf{E}}_l = \max_l p(\mathbf{A}_l|\delta_l, \mathbf{E}_l). \tag{6}$$

After taking the logarithm, (6) becomes

$$\bar{\mathbf{E}}_l = \max_l \left\{ \sum_{k=1}^N \log p\left(\mathbf{A}_{l_k}|\delta_{l_k}, \mathbf{E}_{l_k}\right) \right\} \tag{7}$$

where $p(\mathbf{A}_{l_k}|\delta_{l_k}, \mathbf{E}_{l_k})$ means the observation likelihood of patch $k$ in $l$. Let $p(\mathbf{A}_{l_k}|\delta_{l_k}, \mathbf{E}_{l_k}) \propto e^{\omega_{l_k} v_{l_{kk}}}$. Then, our objective function is finally defined as

$$\bar{\mathbf{E}}_l = \max_l \left\{ \sum_{k=1}^N \omega_{l_k} v_{l_{kk}} \right\} \tag{8}$$

where $\omega_{l_k} = 1 + \delta_{l_k} e^{-\beta(|i-((1+r)/2)|+|j-((1+c)/2)|)}$ denotes the contribution factor of patch $k$ in $l$ and $v_{l_{kk}}$ means the score of patch $k$ in $l$.

### B. Template Update

It is necessary to update the templates in tracking, because fixed templates cannot capture the appearance change of the target. IVT [15] is proposed to update both eigenbasis and mean to faithfully model the appearance change of the target. Although IVT is robust to illumination and pose variation, it is sensitive to partial occlusion. The template update method in [18], which combines subspace learning with sparse representation, is proposed to deal with partial occlusion

$$\mathbf{p} = \mathbf{Uq} + \mathbf{e} = [\mathbf{U}\ \mathbf{I}][\mathbf{q}\ \mathbf{e}]^\top \tag{9}$$

where $\mathbf{p}$ denotes the observation vector, $\mathbf{U}$ is the matrix composed by eigenbasis vectors, $\mathbf{q}$ is the coefficient of eigenbasis vectors, and $\mathbf{e}$ indicates the pixels in $\mathbf{p}$ that are occluded. Let $\hat{\mathbf{U}} = [\mathbf{U}\ \ \mathbf{I}]$ and $\hat{\mathbf{q}} = [\mathbf{q}\ \ \mathbf{e}]^\top$, assuming the error caused by occlusion is sparse, (9) can be solved by

$$\min_{\hat{\mathbf{q}}} \quad \left\|\mathbf{p} - \hat{\mathbf{U}}\hat{\mathbf{q}}\right\|_2^2 + \lambda\left\|\hat{\mathbf{q}}\right\|_1 \tag{10}$$

where $\lambda$ is the regularization parameter. The goal of (10) is to update $\mathbf{Uq}$ into the template set.

As the number of templates is fixed, old template sets need to discarded low-weight templates for balance. Usually, the selection of the discarded template is complied with the rationale that the earlier tracking results are more accurate and should be stored longer than the latter tracking results. In detail, a cumulative probability sequence $\{0, (1/(2^{n-1}-1)), (3/(2^{n-1}-1)), (7/(2^{n-1}-1)), \ldots, 1\}$ is first generated and its each element means the update probability from the first template to the $n$th template. Then, a random number is generated according to the uniform distribution on the unit interval [0, 1]. According to the random number, the corresponding template is discarded.

To some extent, sparse representation can avoid that the shelter is updated into template set. However, the shelter could possibly be updated into template set when the target suffers from a large occlusion. In view of this, we introduce occlusion ratio $r_{\text{occ}}$ to describe the occlusion degree, which is denoted as

$$r_{\text{occ}} = \frac{N - \sum_{i=1}^N \delta_i}{N}. \tag{11}$$

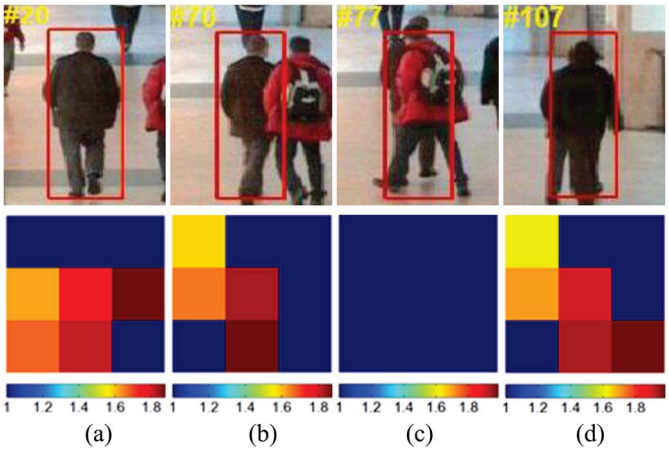



Fig. 4. Illustration of contribution factor design under four cases. The first row means the tracking target and the second row corresponds to the contribution factor of the first row. Note that the patches with different colors have different contribution factors. The red patch stands for the large contribution factor, while the blue patch stands for the small contribution factor. In detail, this figure includes four columns, each one of which includes the upper tracking target figure and the lower contribution factor figure. (a) No occlusion. (b) Partial occlusion. (c) Complete occlusion. (d) Disruptor occlusion.

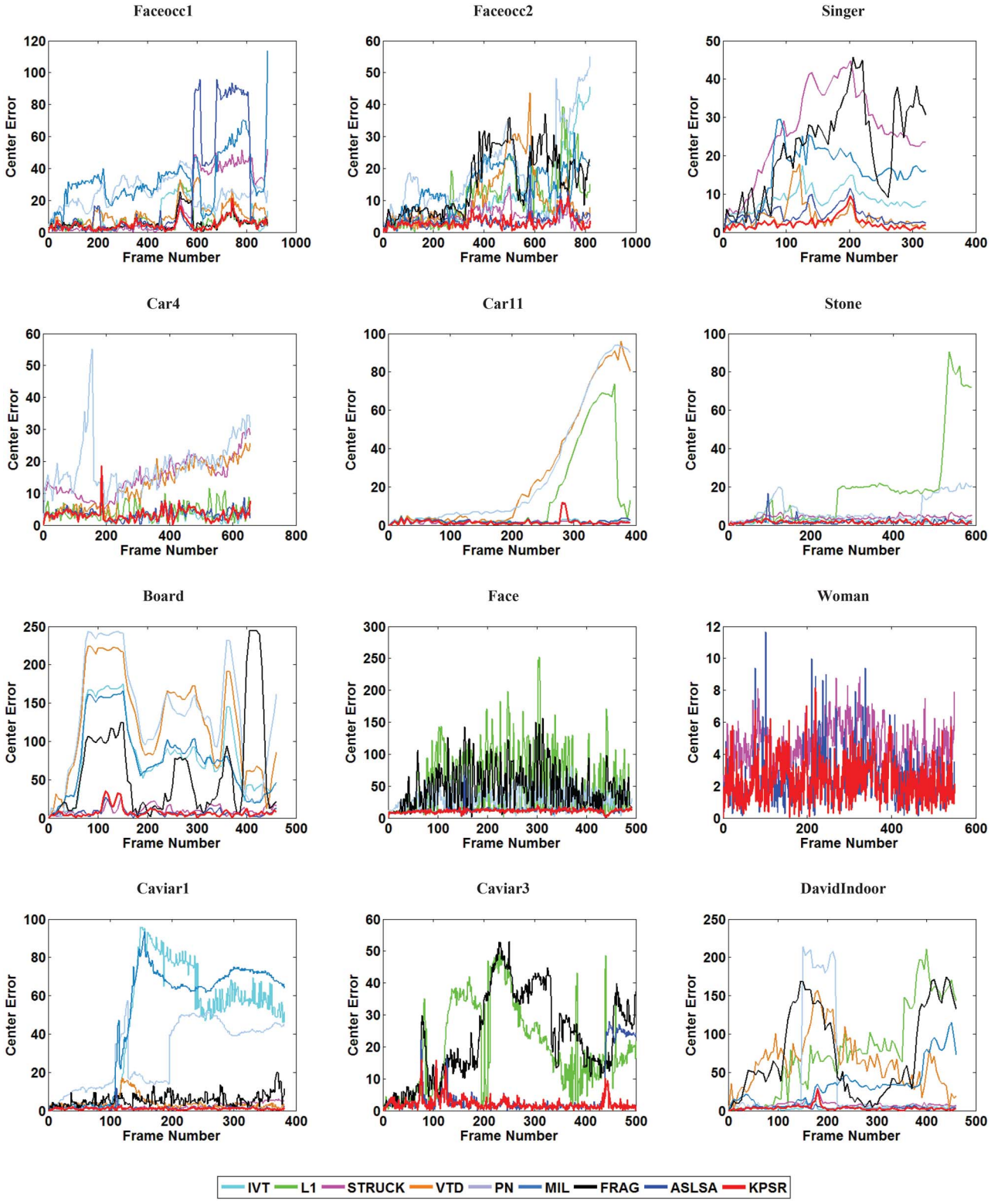



Fig. 5. Quantitative comparisons between KPSR and eight trackers in terms of position CE (in pixels).

We regard $r_{\text{occ}} \leq \eta$ as a constraint to determine wether the template is updated or not. Algorithm 1 gives the procedure of template selection.

### C. Discussion

In this section, we visualize the contribution factor of each patch under three cases, namely the case without occlusion, with partial occlusion, and with complete occlusion, as shown in Fig. 4.

*1) Case Without Occlusion:* Under this case, in theory, we have $\delta_k = 1$, $\omega_k = 1 + e^{-\beta(|i-((1+r)/2)|+|j-((1+c)/2)|)}$, $k = 1, 2, \ldots, N$, and $r_{\text{occ}} = 0$. Therefore, the prediction result should be in a "+" style theoretically. Experimentally, from Fig. 4(a), we can see that our method only predict the second

TABLE I
EXPERIMENTAL COMPARISON RESULTS IN TERMS OF AVERAGE POSITION CES

| | IVT | $\ell 1$ | STRUCK | VTD | PN | MIL | FRAG | ASLSA | KPSR |
|---|---|---|---|---|---|---|---|---|---|
| Faceocc1 | 8.83 | 6.50 | 16.35 | 11.13 | 24.10 | 32.26 | 5.62 | 23.25 | **4.78** |
| Faceocc2 | 10.21 | 11.12 | 5.01 | 10.41 | 18.59 | 14.06 | 15.50 | 3.71 | **3.12** |
| Singer | 8.48 | 105.26 | 26.30 | 4.06 | 121.49 | 15.17 | 22.03 | 4.50 | **2.40** |
| Car4 | **2.87** | 4.08 | 14.74 | 12.29 | 18.73 | 60.10 | 179.77 | 3.56 | 3.30 |
| Car11 | 2.11 | 14.61 | **1.30** | 27.05 | 27.92 | 43.47 | 63.92 | 2.09 | 1.53 |
| Stone | 2.25 | 19.23 | 3.99 | 31.36 | 8.00 | 32.30 | 65.89 | 1.72 | **1.57** |
| Board | 80.07 | 262.71 | 9.89 | 122.62 | 133.08 | 77.33 | 60.76 | 7.57 | **7.36** |
| Woman | 222.16 | 114.18 | 4.62 | 72.44 | 225.78 | 123.33 | 121.75 | 2.57 | **2.33** |
| Face | 14.83 | 53.31 | 101.47 | 151.42 | 32.81 | 108.76 | 48.95 | 11.53 | **11.16** |
| Caviar1 | 45.25 | 119.93 | 2.28 | 3.91 | 29.77 | 48.50 | 5.70 | 1.30 | **1.07** |
| Caviar2 | 8.64 | 3.24 | 6.26 | 4.72 | 8.51 | 70.27 | 5.57 | 1.85 | **1.35** |
| Caviar3 | 66.98 | 19.70 | 69.04 | 118.42 | 82.44 | 107.81 | 23.02 | 4.99 | **2.21** |
| DavidIndoor | 4.18 | 76.66 | 7.19 | 63.36 | 33.84 | 32.94 | 76.14 | 3.56 | **3.49** |

TABLE II
EXPERIMENTAL COMPARISON RESULTS IN TERMS OF AVERAGE OVERLAPPING RATE CRITERION

| | IVT | $\ell 1$ | STRUCK | VTD | PN | MIL | FRAG | ASLSA | KPSR |
|---|---|---|---|---|---|---|---|---|---|
| Faceocc1 | 0.86 | 0.88 | 0.80 | 0.77 | 0.62 | 0.59 | 0.90 | 0.74 | **0.91** |
| Faceocc2 | 0.59 | 0.67 | 0.71 | 0.59 | 0.49 | 0.61 | 0.60 | 0.83 | **0.83** |
| Singer | 0.66 | 0.20 | 0.31 | 0.79 | 0.46 | 0.34 | 0.34 | 0.82 | **0.85** |
| Car4 | **0.92** | 0.84 | 0.49 | 0.73 | 0.64 | 0.34 | 0.22 | 0.90 | 0.90 |
| Car11 | 0.81 | 0.60 | 0.81 | 0.43 | 0.40 | 0.17 | 0.09 | 0.82 | **0.85** |
| Stone | 0.66 | 0.29 | 0.48 | 0.42 | 0.41 | 0.32 | 0.15 | 0.63 | **0.67** |
| Board | 0.32 | 0.09 | **0.87** | 0.23 | 0.23 | 0.42 | 0.56 | 0.74 | 0.70 |
| Woman | 0.19 | 0.13 | 0.71 | 0.21 | 0.03 | 0.17 | 0.17 | 0.81 | **0.82** |
| Face | 0.71 | 0.43 | 0.21 | 0.07 | 0.51 | 0.23 | 0.39 | 0.79 | **0.79** |
| Caviar1 | 0.28 | 0.28 | 0.76 | 0.83 | 0.27 | 0.25 | 0.68 | 0.90 | **0.91** |
| Caviar2 | 0.45 | 0.81 | 0.50 | 0.67 | 0.66 | 0.26 | 0.56 | 0.84 | **0.88** |
| Caviar3 | 0.14 | 0.30 | 0.14 | 0.09 | 0.12 | 0.13 | 0.29 | 0.75 | **0.84** |
| DavidIndoor | 0.70 | 0.20 | 0.54 | 0.09 | 0.57 | 0.31 | 0.19 | **0.79** | 0.57 |

patch wrong (corresponding to the head part of the target). Unlike ASLSA, our tracker differently treats the patch according to its location. Therefore, our tracker can restrain the importance of the nonkey patch while ASLSA cannot.

*2) Case With Partial Occlusion:* Under this case, in theory, we have different values of $\delta_k$, $\omega_k = 1 + \delta_k e^{-\beta(|i-((1+r)/2)|+|j-((1+c)/2)|)}, k = 1, 2, \ldots, N$, and $0 < r_{\text{occ}} < 1$. In this case, we believe that the unoccluded patch is key patch (i.e., $\delta_k = 1$), and should give a larger contribution factor for the key patch. Taking Fig. 4(b) for example, we can see that the right part of the man with black clothes is occluded by the red man. Therefore, in this case, the theoretical prediction result should be in a "⊣" style. Experimentally, from the lower figure of Fig. 4(b), we can see that our method only predict the second patch wrong (corresponding to the head part of the target). Therefore, our tracker can better deal with partial occlusion than ASLSA to some extent.

*3) Case With Complete Occlusion:* Under this case, in theory, we have $\delta_k = 0$, $\omega_k = 1, k = 1, 2, \ldots, N$, and $r_{\text{occ}} = 1$. $r_{\text{occ}} = 1$ indicates that each patch is occluded. From the upper figure of Fig. 4(c), we can see that the red man occlude the target completely. In this case, the theoretical prediction result should be that all patches are occluded. Obviously, our method predict them with a higher accuracy from the lower figure of Fig. 4(c). Considering the occluded target is not supposed to update into template set, our tracker is able to reject the occluded target to update into the template set while ASLSA cannot. Therefore, our tracker may have a better tracking than ASLSA.

All the above three cases are involving a greater distinction between the target and the shelter. In these cases, our tracker can obtain a more robust and stable results than ASLSA in most of cases (see Section V). However, when the target is occluded by a similar disruptor (i.e., the shelter is very similar with the target), our key patch prediction scheme will be invalid [see Fig. 4(d)].

## V. EXPERIMENTS

### A. Implementation Details and Data Sets

Our tracker based on KPSR is implemented in MATLAB and runs at around 4.4 frames/s on a PC with an Intel 3.6 GHz Dual Core CPU and 4 GB memory, which is slower than ASLSA (≈8 frames/s) while faster than $\ell_1$ tracker (≈2 frames/s). In the experiments, we manually label the location of the target in the first frame for each data set and set $\lambda = 0.01$, $n = 15$, $N = 9$, $M = 600$, $\eta = 0.4$, $\beta = 1$, and $\theta = 13$.

We evaluate the performance of KPSR on 13 challenging data sets, which include Faceocc1, Faceocc2, Singer, Car4, Car11, Stone, Board, Woman, Face, Caviar1, Caviar2, Caviar3, and DavidIndoor. For each data set, we resize the bounding box to $32 \times 32$ pixels and sample patches with the patch size $16 \times 16$ pixels and the step length 8 pixels.

### B. Evaluation

We compare KPSR with eight tracking methods, i.e., IVT method [15], $\ell_1$ tracker [16], STRUCK [8], visual tracking

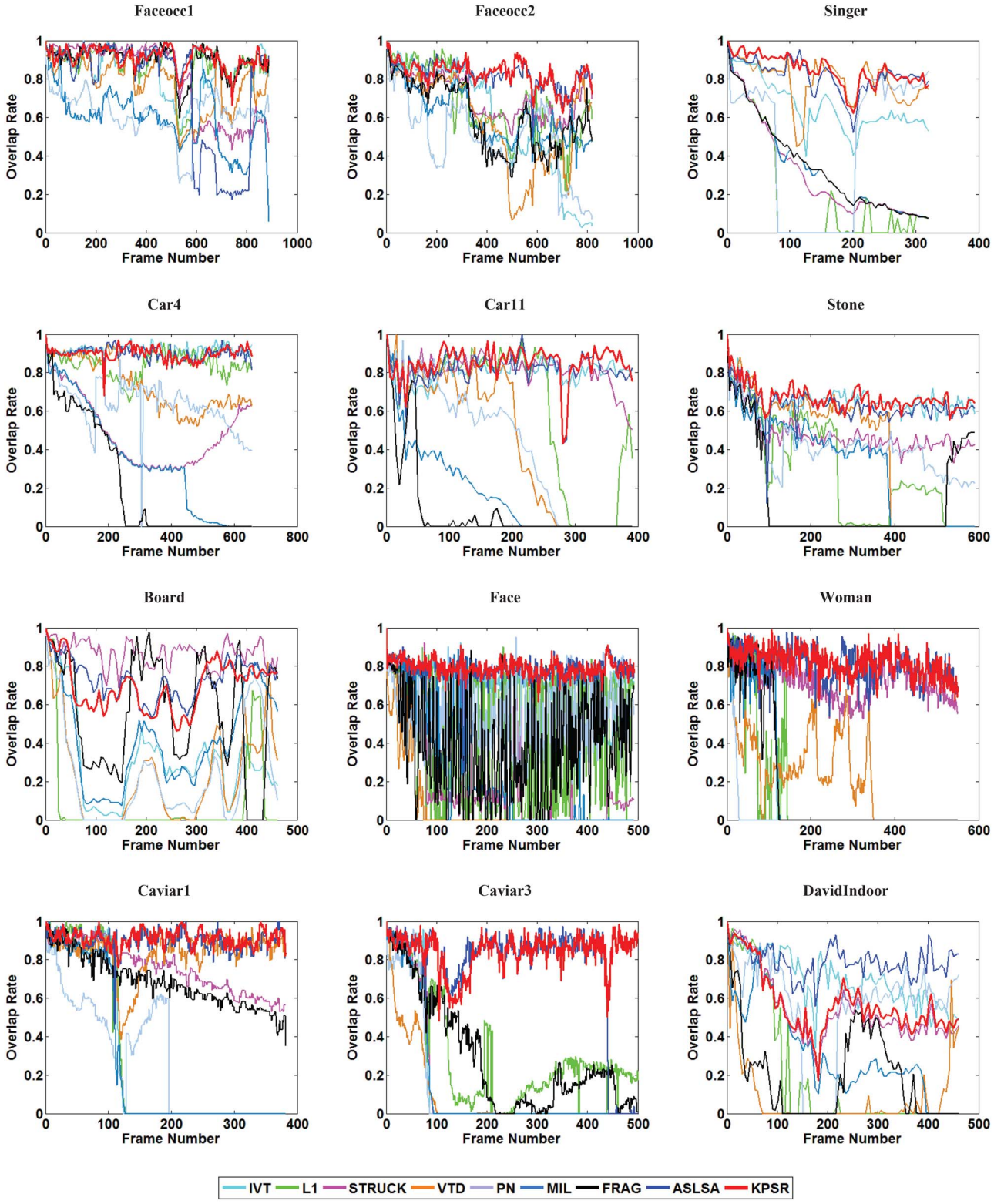



Fig. 6. Quantitative comparisons between KPSR and eight trackers in terms of average overlap errors.

decomposition (VTD) method [13], PN tracker [3], MIL tracker [2], FRAG [11], and ASLSA [18]. IVT, $\ell_1$, MIL, and FRAG methods are classical and the remaining ones are state-of-the-art. In order to make the comparisons objective and persuasive, we obtain the results of these tracking methods by running the source codes provided by their authors.

*1) Quantitative Evaluation:* We employ two evaluation criteria, i.e., the position center error (CE) (in pixels) and the Pascal visual object classes (Pascal VOC) overlap criterion [18], [28], to quantitatively evaluate the performance of KPSR. In detail, given the tracked target and its ground truth, overlap rate (OR) and CE can be denoted as

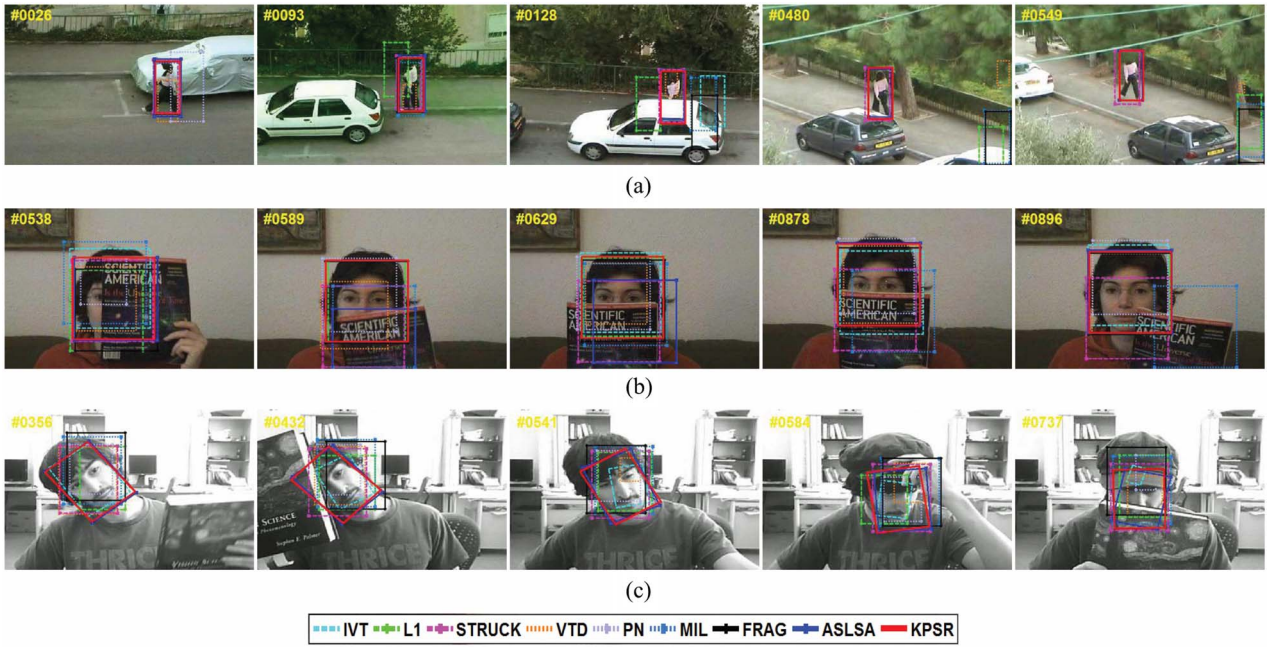



Fig. 7. Tracking results on video data sets with heavy or long-time partial occlusion. (a) Woman. (b) Faceocc1. (c) Faceocc2.

OR $= ((R_T \cap R_G)/(R_T \cup R_G))$ and CE $=$ norm$(C_T - C_G)$, respectively. Here, $R_T$ and $C_T$ are used to represent the region and the center of the tracked target, and $R_G$ and $C_G$ are used to represent the region and the center of the ground truth. It should be noted that only CE or OR cannot ensure the accuracy of the tracking result. And we believe both CE and OR should be used for tracking evaluation. Fig. 5 shows the comparison results between eight tracking methods and our method using the position CE criterion and the average position CE is listed in Table I. For greater clarity, Fig. 5 removes the results of some tracking methods with large CE. Taking Car4 data set for example, we remove two curve lines of MIL and FRAG. This is because these two methods deviate from the correct location after a few frames. Similarly, we remove STRUCK, VTD, and MIL methods on Face data set, IVT, $\ell_1$, PN, MIL, FRAG, and VTD methods on Woman data set, $\ell_1$ method on Caviar1 data set, VTD, MIL, IVT, STRUCK, and PN methods on Caviar3 data set, MIL and FRAG methods on Car11 data set, VTD, MIL, and FRAG methods on Stone data set, $\ell_1$ method on Board data set, and remove $\ell_1$ and PN methods on Singer data set. Fig. 6 shows the comparison results between eight tracking methods and our method using the Pascal VOC overlap criterion and the average overlap error is listed in Table II. From these figures and tables, we can clearly see that KPSR is the best method on Faceocc1, Faceocc2, Singer, Stone, Woman, Face, Caviar1, Caviar2, and Caviar3 data sets. KPSR and ASLSA yield the similar results on DavidIndoor data set, KPSR and IVT yield the similar results on Car4 data set, and KPSR and STRUCK also yield the similar results on Car11 and Board data sets.

*2) Qualitative Evaluation:*

*a) Occlusion case:* Fig. 7 demonstrates how accurate and robust KPSR perform when the target undergoes a heavy or long-time partial occlusion. In the Woman data set, the woman is occluded when she passes by the black car window. MIL, IVT, and FRAG suffer from a drift in this case because the color of the car window is extremely similar with that of the trousers of the woman. However, KPSR, STRUCK, and ASLSA are more robust and stable through the whole data set, this is because KPSR can solve it by increasing the contribution factor for the key patch, STRUCK method benefits robustness to noise by using a kernelized structured output SVM and ASLSA can reduce the partial occlusion to some extent by patch sampling and sparse representation. In the Faceocc1 data set, the human's face is occluded by one book. ASLSA fails to make a correct tracking during the 538th frame to the 878th frame, this is because the occlusion caused by this book not only occupies a large region but also lasts for a long time. However, KPSR is able to make a better tracking, which attributes to occlusion prediction scheme and the adding constraint of $r_{\text{occ}} \leq \eta$ in template update. In the Faceocc2 data set, both KPSR and ASLSA can work well, this is because the occlusion have a short time and particle filter functions it.

*b) Illumination change case:* Fig. 8 demonstrates how accurate and robust KPSR performs when the target undergoes a large illumination variation. More specific, in the Singer data set, many methods suffer from a drift when the target undergoes a large illumination variation. For example, the PN method drifts away in the 142nd frame and recovers a correct tracking due to its global search function. $\ell_1$ also drifts away when the target suffers from the heavy illumination, this is because template update in $\ell_1$ cannot capture the appearance variation of the target. However, KPSR, ASLSA, and IVT have a more robust tracking. This is because the template updates in both KPSR and ASLSA use IVT method and hence are robust to illumination. In the DavidIndoor data set, KPSR and ASLSA have the approximated tracking result. In the Car11 data set, KPSR, IVT, STRUCK, and ASLSA can estimate the more accurate location of the target while the remaining methods fail to estimate the correct location.

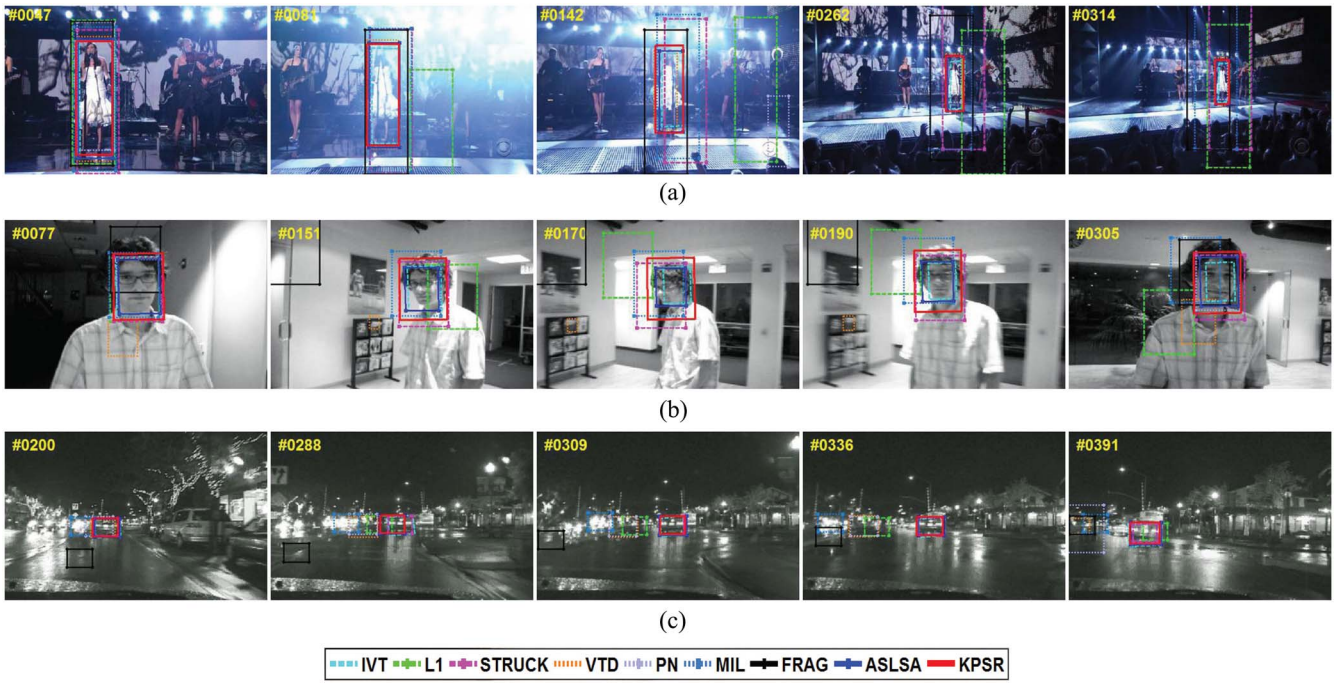



Fig. 8. Tracking results on video data sets with illumination change. (a) Singer. (b) DavidIndoor. (c) Car11.

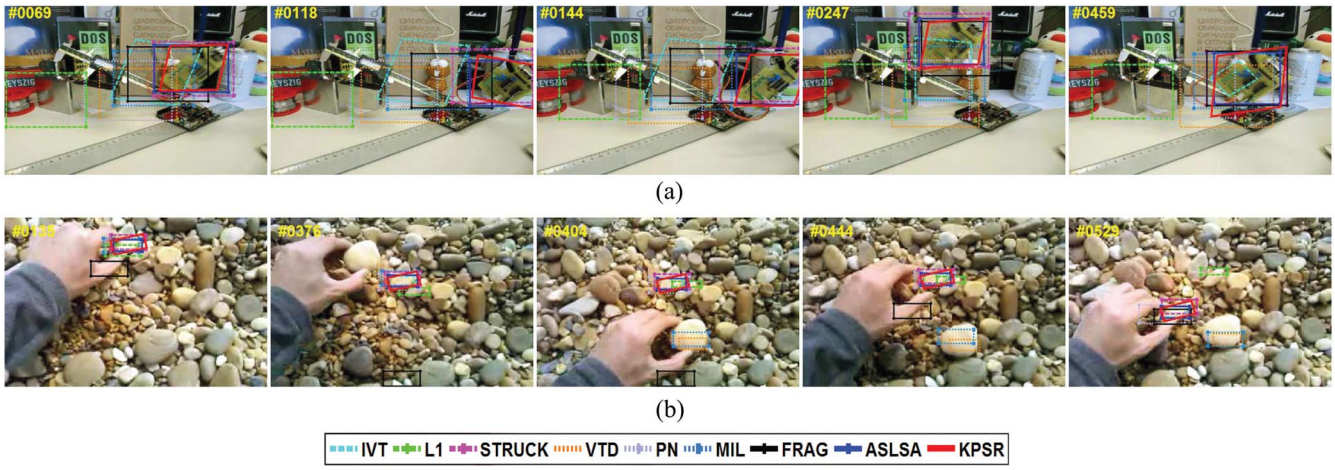



Fig. 9. Tracking results on video data sets with background clutter. (a) Board. (b) Stone.

*c) Background clutter case:* Fig. 9 demonstrates how accurate and robust KPSR performs when the target undergoes a heavy background clutter. More specific, in the Board data set, KPSR is able to obtain a better tracking than ASLSA. This is because the bounding box includes amounts of background information and our contribution factor design helps to reduce the disturbance of the background clutter. Besides, ASLSA also keeps a more continuous tracking through the whole data set. In the Stone data set, when a similar big stone fully occludes the target (the small stone), KPSR can keep the minimal tracking error in a continuous tracking as shown in Fig. 5, while FRAG, MIL, and VTD suffer from an extreme influence since the 376th frame.

## VI. Conclusion

In this paper, we propose a robust tracker based on KPSR. To better solve partial occlusion and background clutter problems, KPSR treats differently the sampled patch according to its location and occlusion case. A contribution factor design for all sampled patches utilizes a weighted approach to important patches. In order to obtain the occlusion case of each patch, we provide an occlusion prediction scheme by training a classifier. In addition, the occlusion degree $r_{\text{occ}}$ is used for template update as a update condition. The experiments on challenging data sets demonstrate that the KPSR tracker not only is accurate and robust for occlusion and background clutter but also is effective for illumination change.

## References

[1] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, 2006, pp. 260–267.

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

[3] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 49–56.
[4] S. Wang, H. Lu, F. Yang, and M.-H. Yang, "Superpixel tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 1323–1330.
[5] F. Yang, H. Lu, and M.-H. Yang, "Robust visual tracking via multiple kernel boosting with affinity constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 242–254, Feb. 2014.
[6] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2002, pp. 561–568.
[7] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[8] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 263–270.
[9] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 63–84, 1998.
[10] 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.
[11] 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.*, New York, NY, USA, 2006, pp. 798–805.
[12] 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.*, Providence, RI, USA, 2011, pp. 1313–1320.
[13] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 1269–1276.
[14] L. 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.
[15] 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.
[16] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proc. Int. Conf. Comput. Vis.*, Kyoto, Japan, 2009, pp. 1436–1443.
[17] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 1257–1264.
[18] 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.*, Providence, RI, USA, 2012, pp. 1822–1829.
[19] B. Liu *et al.*, "Robust and fast collaborative tracking with two stage sparse optimization," in *Proc. IEEE Conf. Comput. Vis.*, Heraklion, Greece, 2010, pp. 624–637.
[20] Y. Wu, E. Blasch, G. Chen, L. Bai, and H. Ling, "Multiple source data fusion via sparse representation for robust visual tracking," in *Proc. 14th Int. Conf. Inf. Fusion (FUSION)*, Chicago, IL, USA, 2011, pp. 1–8.
[21] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.
[22] 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.
[23] Q. Shi, A. Eriksson, A. Van Den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 553–560.
[24] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1994.
[26] B. Efron, R. Tibshirani, T. Hastie, and I. Johnstone, "Least angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.
[27] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, May 2015.
[28] 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, 2010.

**Zhenyu He** (SM'12) received the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2007.

He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. His current research interests include sparse representation and its applications, deep learning and its applications, pattern recognition, image processing, and computer vision.

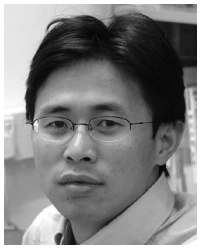
**Shuangyan Yi** received the M.S. degree from the Department of Mathematics, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China, where she is currently pursuing the Ph.D. degree in computer science and technology.

Her current research interests include object tracking, pattern recognition, and machine learning.

**Yiu-Ming Cheung** (SM'06) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong.

He is a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include artificial intelligence, visual computing, and optimization.

Prof. Cheung is the Founding and Past Chairman of the Computational Intelligence Chapter of the IEEE Hong Kong Section. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Knowledge and Information Systems*, and the *International Journal of Pattern Recognition and Artificial Intelligence*. He is a Senior Member of ACM.



**Xinge You** (M'08–SM'10) received the B.S. and M.S. degrees in mathematics from Hubei University, Wuhan, China, in 1990 and 2000, respectively, and the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2004.

He is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan. His current research interests include wavelets and its application, signal and image processing, pattern recognition, machine learning, and computer vision.

**Yuan Yan Tang** (F'04) received the B.S. degree in electrical and computer engineering from Chongqing University, Chongqing, China, the M.Eng. degree in electrical engineering from the Beijing Institute of Post and Telecommunications, Beijing, China, and the Ph.D. degree in computer science from Concordia University, Montreal, QC, Canada.

He is currently a Chair Professor with the Faculty of Science and Technology, University of Macau, Macau, China, and a Professor/Adjunct Professor/Honorary Professor with several institutes, including several universities in China, Concordia University, Canada, and Hong Kong Baptist University, Hong Kong. He has published over 400 technical papers and authored/co-authored over 25 monographs/books/bookchapters on subjects ranging from electrical engineering to computer science. His current research interests include wavelet theory and applications, pattern recognition, image processing, document processing, artificial intelligence, and Chinese computing.

Dr. Tang is the Founder and the Editor-in-Chief of the *International Journal on Wavelets*, *Multiresolution*, and *Information Processing*, and an Associate Editor of several international journals, such as the *International Journal on Pattern Recognition and Artificial Intelligence*. He is the Founder and the Chair of Pattern Recognition Committee in the IEEE SYSTEMS, MAN, AND CYBERNETICS. He has served as the General Chair, the Program Chair, and a Committee Member for many international conferences, including the General Chair of the 18th International Conference on Pattern Recognition. He is the Founder and the General Chair of the series International Conferences on Wavelets Analysis and Pattern Recognition. He is a fellow of the International Associate of Pattern Recognition.