Visual Tracking via Probability Continuous Outlier Model

Dong Wang
Dalian University of Technology, Dalian, China
wdice@dlut.edu.cn
wangdong.ice@gmail.com

Huchuan Lu
Dalian University of Technology, Dalian, China
lhchuan@dlut.edu.cn
luhuchuan@gmail.com

Abstract

In this paper, we present a novel online visual tracking method based on linear representation. First, we present a novel probability continuous outlier model (PCOM) to depict the continuous outliers that occur in the linear representation model. In the proposed model, the element of the noisy observation sample can be either represented by a PCA subspace with small Guassian noise or treated as an arbitrary value with a uniform prior, in which the spatial consistency prior is exploited by using a binary Markov random field model. Then, we derive the objective function of the PCOM method, the solution of which can be iteratively obtained by the outlier-free least squares and standard max-flow/min-cut steps. Finally, based on the proposed PCOM method, we design an effective observation likelihood function and a simple update scheme for visual tracking. Both qualitative and quantitative evaluations demonstrate that our tracker achieves very favorable performance in terms of both accuracy and speed.

1. Introduction

As one of the fundamental topics in computer vision, visual tracking plays a key role in numerous lines of research and has many practical applications such as video surveillance, human computer interaction, traffic control, motion analysis, activity analysis, driver assistance system and so on. While much work has been done [26] in the past decades, designing a robust tracking algorithm remains a challenging task due to numerous factors including illumination variation, partial occlusion, pose change, motion blur, background clutter, and many more. A typical tracking system includes two basic components: (1) a motion model, which relates the states of an object over time and supplies the tracker with a number of candidate states (e.g., Kalman filter [6], particle filter [16]) ; (2) an observation model, which represents the tracked object and evaluates the likelihood of each candidate state in the current frame. In this paper, we focus on developing an effective observation model due to its crucial role for visual tracking.

Existing observation models can be categorized into methods based on templates (e.g., [1, 14]), online classifiers (e.g., [2, 12]), linear representation models (e.g., [19, 18, 24, 23]) and so on. In the template-based algorithms, the tracked object is described by one single template [6] or multiple templates [14]. Then the tracking problem can be considered as searching for the regions which are the most similar to the tracked object. The trackers based on online classifiers treat tracking as a binary classification problem, which aims to distinguish the tracked targets from its surrounding backgrounds. Both classic and recent machine learning algorithms could promote the progress of tracking algorithms or systems, including boosting [9], support vector machines [21], naive bayes [27], random forests [20], multiple instance learning [2], matting [7] and so on. In this work, we propose a fast and effective generative tracker based on the linear representation model, which is able to deal with continuous outliers and therefore provides an accurate match effectively.

Instead of representing the tracked object as a collection of low-level features, the linear representation models maintain holistic appearance information and therefore provide a compact notion of the “thing” being tracked [19, 18]. Ross *et al.* [19] propose an incremental visual tracking (IVT) method, which represents the tracked object by using a low dimensional principle component analysis (PCA) [22] subspace and assumes that the representation error is Gaussian distributed with small variances. Although the IVT method is robust to illumination and pose changes, it is very sensitive to partial occlusion and background clutter. The underlying reason is the noise term cannot be modeled with small variances when some outlier occurs.

Inspired by the idea of sparse representation [25], Mei *et al.* develop a novel ℓ_1 tracker [18], which uses a series of object and trivial templates to represent the tracked object. In the ℓ_1 tracker, object templates are used to describe the object class to be tracked and trivial templates are adopted to deal with outliers (e.g., partial occlusion) with sparsity constraints. Furthermore, several methods improve the original ℓ_1 tracker in terms of both speed and accuracy by using accelerated proximal gradient algorithms [3], replacing raw pixel templates with orthogonal basis vectors [24], modeling the similarity between different candidates [28], to name a few. Although these algorithms explicitly consider outliers by introducing additional trivial templates, they lack of some theoretical foundation and fail to consider the spatial information among outliers.

This paper presents a novel effective and fast tracking method with an adaptive observation model, the main contributions of which are three-folds. First, we represent the tracked object with the linear representation model and the proposed probability continuous outlier model (PCOM), which exploits the spatial information among outliers by using a first-order Markov field. Second, we propose an iteration algorithm to solve the representation coefficient and infer outliers simultaneously. Finally, we develop a generative tracker based on our PCOM method and a simple update scheme. By using twelve challenging video clips, numerous experiments are conducted to illustrate both effectiveness and efficiency of the proposed method.

2. Background and Related Work

2.1. Object tracking via incremental visual tracking (IVT)

Tracking algorithms based on the linear representation model have attracted much attention in recent years (e.g., [19, 18, 10, 24]). Among these methods, the most influential one is the incremental visual tracking (IVT) method [19]. The IVT method introduces an online update approach for efficiently learning and updating a low dimensional PCA subspace representation of the target object. Several experimental results show this method is effective in dealing with appearance change caused by in-plane rotation, illumination variation and pose change. But this method is very sensitive to partial occlusion, which can be explained by equation (1).

$$\mathbf{y} = \mathbf{U}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{y}$ denotes an observation vector, $\mathbf{x}$ indicates the coefficient vector, $\mathbf{U}$ represents a matrix of column basis vectors, and $\mathbf{e}$ stands for the error vector.

In the PCA representation model, the error term $\mathbf{e}$ is assumed to be Gaussian distributed with small variances. Based on the maximum likelihood estimation, the coefficient vector $\mathbf{x}$ can be estimated by $\mathbf{x} = \mathbf{U}^\top\mathbf{y}$, and the reconstruction error can be approximated by $\left\|\mathbf{y} - \mathbf{U}\mathbf{U}^\top\mathbf{y}\right\|_2^2$. However, this assumption does not hold when outliers occur as outliers cannot be modeled by the Gaussian distribution with small variance. Hence, the IVT method is sensitive to partial occlusion and background clutter. Although some recent works [18, 24] explicitly consider outliers by treating the error term $\mathbf{e}$ as arbitrary but sparse noise, they lack of some theoretical foundation and fail to consider the spatial information among outliers.

2.2. Graph cuts

The Graph cuts method [4] solves energy minimization problems by constructing a graph and computing the min-cut, which is usually adopted to segment images or videos. Take image segmentation as an instance, the Graph cuts method computes a segmentation over a set of pixels P by the following objective function,

$$E(\mathbf{f}) = \sum_{p_i \in P} R(p_i, f_i) + \lambda \sum_{(p_i, p_j) \in E} B(p_i, p_j)\, |f_i - f_j|, \tag{2}$$

where $\mathbf{f} = \left[f_1, f_2, ..., f_{|P|}\right]^\top$ is a binary vector of labels and f_i indicates the label of the i-th pixel p_i ($f_i = 1$ stands for foreground; and $f_i = 0$ means background). $R(p_i, f_i)$ is a region cost term based on the label that depicts the individual property of a given pixel, and $B(p_i, p_j)$ is a boundary cost term based on the neighbor set E that models the spatial relationship between adjacent pixels. The parameter λ balances the importances of R and B. There exist two important conclusions on the Graph cuts method: first, the objective function (2) can be effectively solved by the max-flow/min-cut method [8, 13]; second, the objective function (2) can be viewed as the energy function of a first-order binary Markov random field [15].

3. Probability Continuous Outlier Model (PCOM)

We present the proposed PCOM method based on the linear representation model, which aims to solve the representation coefficient based on a series of noisy observations, $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is a n-dimensional observation vector, $\mathbf{x} \in \mathbb{R}^{k\times 1}$ is the representation coefficient, and $\mathbf{e} = \mathbf{y} - \mathbf{A}\mathbf{x}$ denotes the error term. In the field of computer vision, the matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_k]$ is usually known as dictionary or basis matrix, where $\mathbf{a}_i$ is called an atom or basis vector.

In this work, we adopt a PCA [19] model (centered at $\boldsymbol{\mu}$, spanned by the orthogonal bases $\mathbf{U}$) to represent an object, in which a given image patch should be firstly converted into one column vector. Thus, we denote $\mathbf{y} \leftarrow \mathbf{y} - \boldsymbol{\mu}$ and $\mathbf{A} = \mathbf{U}$ for simplification. In order to model the continuous outliers explicitly, we introduce a binary indicator vector $\mathbf{w} = [w_1, w_2, ..., w_n]^\top$ to indicate inliers or outliers (i.e., $w_i = 1$ means y_i is an inlier; $w_i = 0$ means y_i is an outlier).

Inlier: If y_i is an inlier (i.e., $w_i = 1$), it can be represented by the linear representation model with small Gaussian noise (i.e., zero-mean Gaussian random variable with variance σ^2). Its conditional probability density function is,

$$p\left(y_i|w_i=1,\mathbf{x}\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\left[y_i-\left(\mathbf{A}\mathbf{x}\right)_i\right]^2\Big/2\sigma^2\right\}. \tag{3}$$

Outlier: If y_i is an outlier (i.e., $w_i=0$), it can be chosen as an arbitrary value in the interval $[a,b]$ ($[a,b]$ depicts the range of image values and has same value for different elements). Thus, we adopt a uniform distribution (equation (4)) to model the conditional probability of an outlier. We note that the probability distribution of an outlier has no relation to the representation coefficient $\mathbf{x}$.

$$p\left(y_i|w_i=0,\mathbf{x}\right)=\left\{\begin{array}{ll}\frac{1}{b-a}, & a\le y_i\le b\\ 0, & otherwise\end{array}\right. \tag{4}$$

Spatial Continuity: The image domain can be treated as a graph $G=(V,E)$, where $V=\{y_1,y_2,...,y_n\}$ denotes the vertex set that consists of n pixels and E stands for the edges connecting neighboring pixels (we use the standard 4-neighbor in this work). The spatial continuity among the outliers (and also the inliers as well) can be modeled by a Markov random field (MRF) [15]. In this work, we adopt a simple *Ising Model* for the probability distribution function of the indicator vector $\mathbf{w}$:

$$p\left(\mathbf{w}\right)=\frac{1}{Z}\exp\left(-\sum\nolimits_{i,j\in E}\beta_{ij}\left|w_i-w_j\right|\right), \tag{5}$$

where β_{ij} controls the interaction between indicator values w_i and w_j, and Z is a normalization constant.

Assuming there is a uniform prior on the coefficient $\mathbf{x}$, the posterior probability $p\left(\mathbf{w},\mathbf{x}|\mathbf{y}\right)$ can be derived as

$$\begin{array}{l}p\left(\mathbf{w},\mathbf{x}|\mathbf{y}\right)\propto p\left(\mathbf{y}|\mathbf{w},\mathbf{x}\right)p\left(\mathbf{w}\right)\\ =\left[\prod\limits_{i=1}^{n}p\left(\mathbf{y}_i|w_i,\mathbf{x}\right)\right]p\left(\mathbf{w}\right)\\ =\left[\prod\limits_{i=1}^{n}p(\mathbf{y}_i|w_i=1,\mathbf{x})^{w_i}p(\mathbf{y}_i|w_i=0,\mathbf{x})^{1-w_i}\right]p\left(\mathbf{w}\right)\\ =\left\{\prod\limits_{i=1}^{n}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y_i-\left(\mathbf{A}\mathbf{x}\right)_i\right)^2}{2\sigma^2}\right)\right]^{w_i}\left(\frac{1}{b-a}\right)^{1-w_i}\right\}\\ \times\frac{1}{Z}\exp\left(-\sum\nolimits_{i,j\in E}\beta_{ij}\left|w_i-w_j\right|\right)\end{array}. \tag{6}$$

The optimal parameters $\widehat{\mathbf{w}}$ and $\widehat{\mathbf{x}}$ can be obtained by maximizing the posteriori probability $p\left(\mathbf{w},\mathbf{x}|\mathbf{y}\right)$, which is equivalent to minimizing the negative logarithm function $-\log p\left(\mathbf{w},\mathbf{x}|\mathbf{y}\right)$. It is not difficult to derive that $-\log p\left(\mathbf{w},\mathbf{x}|\mathbf{y}\right)=C+\frac{1}{\sigma^2}J\left(\mathbf{w},\mathbf{x}\right)$, where C stands for a specific constant and $J\left(\mathbf{w},\mathbf{x}\right)$ is defined as

$$\begin{array}{l}J\left(\mathbf{w},\mathbf{x}\right)=\sum\limits_{i=1}^{n}\left\{w_i\frac{\left[y_i-\left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2}+\left(1-w_i\right)\frac{\lambda^2}{2}\right\}\\ \qquad+\sum\nolimits_{i,j\in E}\lambda_{ij}\left|w_i-w_j\right|\end{array}, \tag{7}$$

where $\lambda=\left(2\sigma^2\log\frac{b-a}{\sqrt{2\pi}\sigma}\right)^{\frac{1}{2}}$ (i.e., $\frac{\lambda^2}{2}=\sigma^2\log\frac{b-a}{\sqrt{2\pi}\sigma}$) and $\lambda_{ij}=\sigma^2\beta_{ij}$ (the detailed derivations can be found in Remark 1). In practice, it does not require to know or estimate the variables σ^2, b, a and β_{ij}. We can choose appropriate regularization parameters λ and λ_{ij} instead, because these parameters have definite physical meanings. The parameter λ can be explained as a threshold to designate inliers and outliers (i.e., if $\left|y_i-\left(\mathbf{A}\widehat{\mathbf{x}}\right)_i\right|\le\lambda$, y_i is indicated as an inlier; otherwise, it is viewed as an outlier). In addition, the parameter λ_{ij} controls the spatial relationship between the indicator vector $\mathbf{w}$.

To the best of our knowledge, there is no close-form solution for the optimization problem (7). Thus, we present an iteration algorithm to compute the optimal parameters $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{w}}$ based on the following propositions.

Proposition 1: If $\widehat{\mathbf{w}}$ is obtained, the optimal $\widehat{\mathbf{x}}$ can be solved by an outlier-free least squares process.

If $\widehat{\mathbf{w}}$ is given, it merely requires to consider the first term in the objective function as the remaining ones are constants. Then the minimization of equation (7) is equivalent to the minimization of

$$F\left(\mathbf{x}\right)=\sum_{\hat{w}_i\neq 0}\frac{1}{2}\left[y_i-\left(\mathbf{A}\mathbf{x}\right)_i\right]^2. \tag{8}$$

Thus, the optimal $\widehat{\mathbf{x}}$ can be obtained by an outlier-free least squares process $\widehat{\mathbf{x}}=\left(\mathbf{A}_*^\top\mathbf{A}_*\right)^{-1}\mathbf{A}_*^\top\mathbf{y}_*$, where $\mathbf{A}_*$ is organized by the rows of $\mathbf{A}$ that are corresponding to the nonzero elements of the indicator vector $\mathbf{w}$ and $\mathbf{y}_*$ is organized in the same manner.

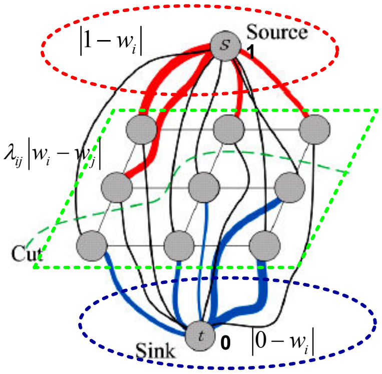


Figure 1. An illustration of Proposition 2.

Proposition 2: If $\widehat{\mathbf{x}}$ is given, the optimal $\widehat{\mathbf{w}}$ can be obtained by the standard max-flow/min-cut algorithm [8] effectively.

If $\widehat{\mathbf{x}}$ is known, the minimization of equation (7) can be converted to the minimization of

$$\begin{array}{l}G\left(\mathbf{w}\right)=\sum\limits_{i=1}^{n}\left(\left|0-w_i\right|\frac{e_i^2}{2}+\left|1-w_i\right|\frac{\lambda^2}{2}\right)\\ \qquad+\sum\limits_{i,j\in E}\lambda_{ij}\left|w_i-w_j\right|\end{array} \tag{9}$$

Remark 1: A detailed derivation from equation (6) to equation (7).
Here we present a detailed derivation from equation (6) to equation (7), i.e., from $-\log p\left(\mathbf{w}, \mathbf{x}|\mathbf{y}\right)$ to $J\left(\mathbf{w}, \mathbf{x}\right)$.

$$\begin{aligned}
&\log p\left(\mathbf{x}, \mathbf{w}|\mathbf{y}\right)\\
&= C_0 + \sum_{i=1}^{n}\left\{-w_i \log\sqrt{2\pi}\sigma - w_i \frac{\left[y_i - \left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2\sigma^2} - \left(1-w_i\right)\log^{b-a}\right\} - \sum_{i,j\in E}\beta_{ij}\left|w_i - w_j\right|\\
&= C_0 + \sum_{i=1}^{n}\left\{-\log\sqrt{2\pi}\sigma - w_i \frac{\left[y_i - \left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2\sigma^2} - \left(1-w_i\right)\left(\log^{b-a} - \log\sqrt{2\pi}\sigma\right)\right\} - \sum_{i,j\in E}\beta_{ij}\left|w_i - w_j\right|\\
&= C + \tfrac{1}{\sigma^2}\left\{-w_i \frac{\left[y_i - \left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2} - \left(1-w_i\right)\sigma^2 \log^{\frac{b-a}{\sqrt{2\pi\sigma}}}\right\} - \sum_{i,j\in E}\beta_{ij}\left|w_i - w_j\right|\\
&= C + \tfrac{1}{\sigma^2}\left\{-w_i \frac{\left[y_i - \left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2} - \left(1-w_i\right)\sigma^2 \log^{\frac{b-a}{\sqrt{2\pi\sigma}}} - \sum_{i,j\in E}\sigma^2\beta_{ij}\left|w_i - w_j\right|\right\}
\end{aligned}$$

where C_0 and C denote some constants to make that the equality holds.
By introducing $\lambda = \left(2\sigma^2 \log^{\frac{b-a}{\sqrt{2\pi\sigma}}}\right)^{\frac{1}{2}}$ (i.e., $\frac{\lambda^2}{2} = \sigma^2 \log^{\frac{b-a}{\sqrt{2\pi\sigma}}}$) and $\lambda_{ij} = \sigma^2\beta_{ij}$, we can obtain that $-\log p\left(\mathbf{w}, \mathbf{x}|\mathbf{y}\right) = C + \frac{1}{\sigma^2} J\left(\mathbf{w}, \mathbf{x}\right)$, where $J\left(\mathbf{w}, \mathbf{x}\right)$ is defined as

$$J\left(\mathbf{w}, \mathbf{x}\right) = \sum_{i=1}^{n}\left\{w_i \frac{\left[y_i - \left(\mathbf{A}\mathbf{x}\right)_i\right]^2}{2} + \left(1-w_i\right)\frac{\lambda^2}{2}\right\} + \sum\nolimits_{i,j\in E}\lambda_{ij}\left|w_i - w_j\right|.$$

by introducing an error term $\mathbf{e} = \mathbf{y} - \mathbf{A}\widehat{\mathbf{x}}$ (i.e., $e_i = y_i - \left(\mathbf{A}\widehat{\mathbf{x}}\right)_i$). The equation (9) can be viewed as the energy function in the Graph Cuts problem [13], and therefore can be minimized by using the max-flow/min-cut algorithm [8] (an intuitive explanation of this proposition is illustrated in Figure 1).

By Propositions 1 and 2, the optimization problem (7) can be solved iteratively. Our empirical results demonstrate that it suffices to use 5 iterations for solving this problem. Figure 2 illustrates the results of some toy examples obtained by the proposed PCOM algorithm, where the red color denotes the inlier mask and the blue color stands for the outlier mask. We can see that the proposed PCOM method is able to estimate inlier-outlier masks accurately.

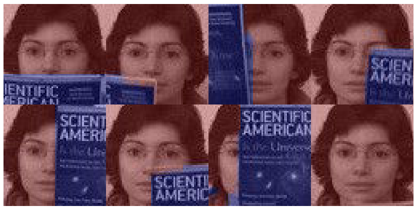


Figure 2. Some toy instances for the proposed PCOM algorithm.

4. The PCOM-based tracking framework

Similar to [19], visual tracking is also cast as a Bayesian inference task with a hidden Markov model. Given a series of observed samples $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_t\}$ up to the t-th frame, the aim is to estimate the hidden state variable $\mathbf{z}_t$ recursively,

$$p\left(\mathbf{z}_t|\mathbf{y}_{1:t}\right) \propto p\left(\mathbf{y}_t|\mathbf{z}_t\right)\int p\left(\mathbf{z}_t|\mathbf{z}_{t-1}\right)p\left(\mathbf{z}_{t-1}|\mathbf{y}_{1:t-1}\right)d\mathbf{z}_{t-1}, \tag{10}$$

where $p\left(\mathbf{z}_t|\mathbf{z}_{t-1}\right)$ stands for the motion model between two consecutive states and $p\left(\mathbf{y}_t|\mathbf{z}_t\right)$ denotes the observation model that estimates the likelihood of an observed image patch belonging to the object class. The flowchart of our tracking framework is illustrated in Figure 3. Similar to the IVT method [19], we use six parameters of the affine transform to depict the motion model $p\left(\mathbf{z}_t|\mathbf{z}_{t-1}\right)$. The state transition is formulated by random walk, i.e., $p\left(\mathbf{z}_t|\mathbf{z}_{t-1}\right) = \mathcal{N}\left(\mathbf{z}_t; \mathbf{z}_{t-1}, \Psi\right)$, where Ψ is a diagonal covariance matrix.

Observation model: By assuming that the variation of the indicator vectors between two consecutive frames is very small, we build our likelihood function based on an outlier-free least squares manner. For each observed image vector corresponding to a predicted state, we solve the following equation by using the least squares algorithm,

$$\widehat{\mathbf{x}}_t^i = \arg\min_{\mathbf{x}_t^i}\left\|\widehat{\mathbf{w}}_{t-1}\odot\left(\mathbf{y}_t^i - \mathbf{A}\mathbf{x}_t^i\right)\right\|_2^2, \tag{11}$$

where i denotes the i-th sample of the state $\mathbf{z}_t$, t is the frame index, and $\odot$ stands for the element-wise multiplication operator. $\widehat{\mathbf{w}}_{t-1}$ is the indicator vector that is obtained based on the PCOM method in frame $t-1$. After the optimal $\widehat{\mathbf{x}}_t^i$

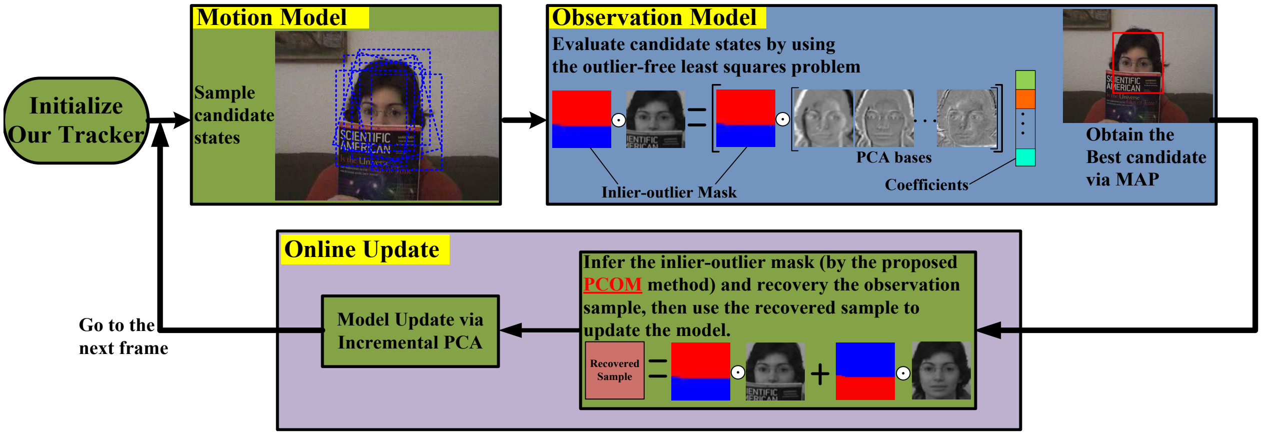


Figure 3. The flowchart of the proposed tracking framework. There exist three fundamental components: motion model; observation model; and online update. This work focuses on the latter two components, especially the observation model.

is obtained, the observation likelihood can be measured by

$$p\left(\mathbf{y}_t^i|\mathbf{z}_t^i\right) \propto \exp\left[-\frac{1}{\gamma}\left\|\widehat{\mathbf{w}}_{t-1}\odot\left(\mathbf{y}_t^i-\mathbf{A}\widehat{\mathbf{x}}_t^i\right)\right\|_2^2\right], \quad (12)$$

where γ is simply set to 0.1 in this work.

Online update: In the proposed PCOM algorithm, the zero components of the indicator vector $\mathbf{w}$ are able to identify outliers. After obtaining the best state of each frame, we extract its corresponding observation vector $\mathbf{y}_o$ and infer the indicator vector $\mathbf{w}_o$. Then we recovery the observation vector by replacing the outliers with its corresponding parts of the mean vector $\boldsymbol{\mu}$,

$$\mathbf{y}_r = \mathbf{w}_o \odot \mathbf{y}_o + (\mathbf{1}-\mathbf{w}_o)\odot\boldsymbol{\mu}, \quad (13)$$

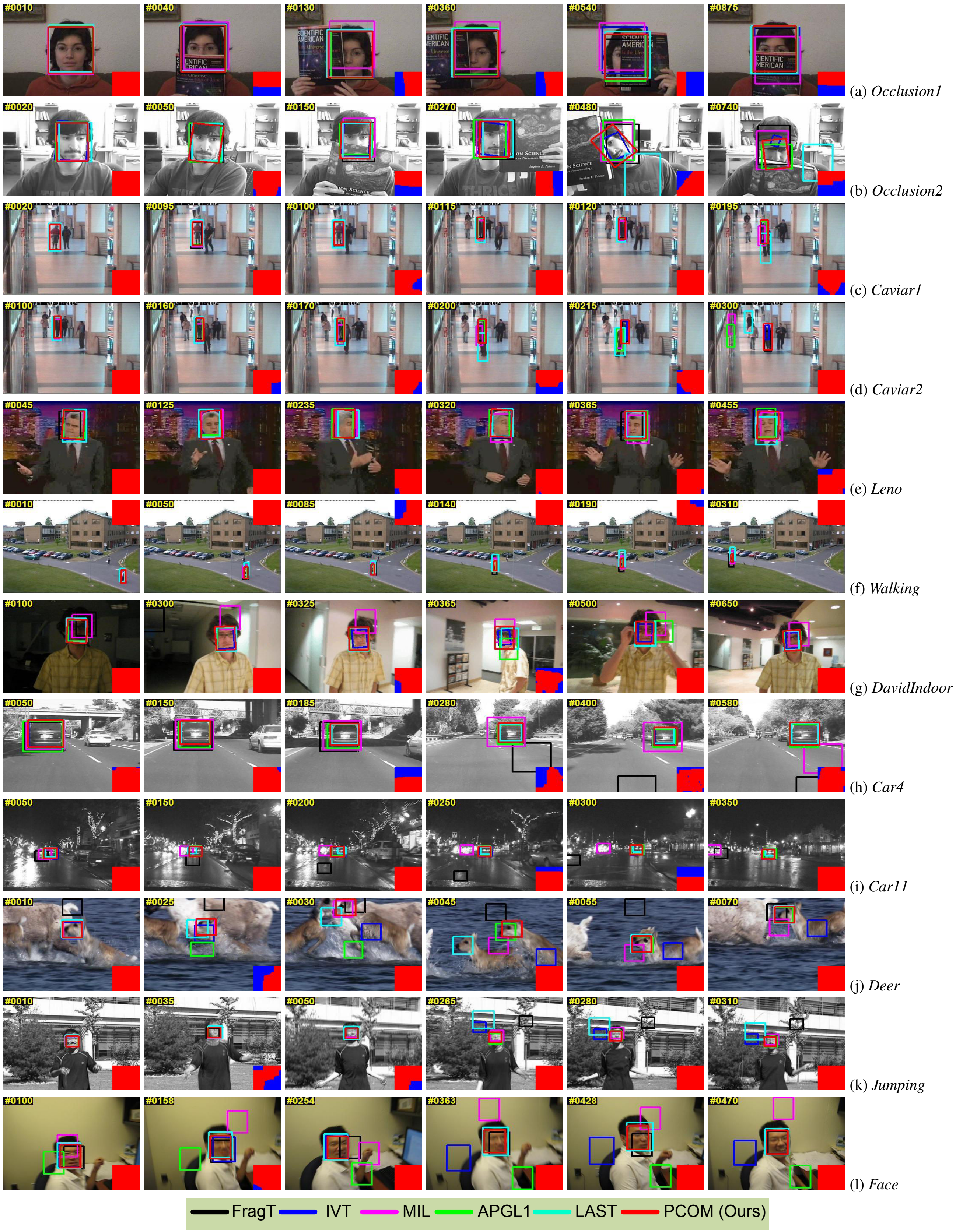
where $\mathbf{y}_r$ denotes the recovered sample and $\odot$ stands for the element-wise multiplication operator. The recovered sample is cumulated and then used to update the tracker via an incremental PCA method [19]. In addition, the inferred indicator vector $\mathbf{w}_o$ is stored, and then used in the next frame (i.e., $\widehat{\mathbf{w}}_t = \mathbf{w}_o$).

5. Experiments

The proposed tracker is implemented in MATLAB 2009B on a PC with Intel i7-3770 CPU (3.4 GHz) with 32 GB memory, and runs 20 frames per second (fps) in this platform. We resize each observation sample to 32×32 pixels and adopt 16 PCA basis vectors. As a trade-off between effectiveness and speed, 600 particles are used and our tracker is incrementally updated every 5 frames. The regularization parameters are set as $\lambda = 0.08$ and $\lambda_{ij} = 0.02$. The MATLAB source codes and datasets are available on our websites (`http://ice.dlut.edu.cn/lu/publications.html`).

In this work, we adopt twelve challenging image sequences from prior work [19, 2, 14] and the CAVIAR data set [5]. The challenges of these sequences include partial occlusion, illumination variation, pose change, background clutter and motion blur. By using these video clips, we evaluate our tracker against ten state-of-the-art tracking algorithms, including the fragment-based tracking (FragT) [1], incremental visual tracking (IVT) [19], multiple instance learning (MIL) [2], visual tracking decomposition (VTD) [14], tracking learning detection (TLD) [12], accelerated proximal gradient L1 (APGL1) [3], local sparse appearance tracking (LSAT) [17], adaptive structural local sparse appearance (ASLSA) [11], multi-task tracking (MTT) [28] and online sparse prototypes tracking (OSPT) [24] methods. For fair evaluation, we use the source codes provided by the authors and run them with adjusted parameters.

Qualitative evaluation: Figure 4 (a)-(b) demonstrate that the proposed tracker performs well in terms of position, scale and rotation when the tracked objects undergo severe occlusion. This can be attributed to two main reasons: (1) the proposed PCOM algorithm takes outliers (e.g., occlusion) and their spatial information into account explicitly. The estimated inlier-outlier masks are able to reflect the occluded or un-occluded portions accurately; (2) the update scheme is able to remove the outliers from new observed samples and therefore avoid degrading the observation model. The FragT [1] method deals with occlusion via the part-based representation, which works well on some simple occlusion cases (e.g., *Occlusion1*). However, this method performs poorly on more challenging cases (e.g., *Occlusion2* and *Caviar2*) since it cannot deal with appearance changes caused by pose and scale. For the same reason, the LSAT [17] method also achieves not good performance when occlusion and other challenging factors oc-



(a) *Occlusion1*

(b) *Occlusion2*

(c) *Caviar1*

(d) *Caviar2*

(e) *Leno*

(f) *Walking*

(g) *DavidIndoor*

(h) *Car4*

(i) *Car11*

(j) *Deer*

(k) *Jumping*

(l) *Face*

Figure 4. Qualitative evaluation of different tracking algorithms on twelve challenging image sequences. The estimated inlier-outlier masks are shown in the lower right (or upper right) of each frame, where the red color stands for inliers and the blue one indicates outliers.

Table 1. Average center location errors of tracking algorithms. The best three results are shown in red, blue and green fonts.

	FragT [1]	IVT [19]	MIL [2]	VTD [14]	TLD [12]	APGL1 [3]	LSAT [17]	ASLAS [11]	MTT [28]	OSPT [24]	Ours
Occlusion1	5.6	9.2	32.3	11.1	17.6	6.8	5.3	10.8	14.1	4.7	5.9
Occlusion2	15.5	10.2	14.1	10.4	18.6	6.3	58.6	3.7	9.2	4.0	4.5
Caviar1	5.7	45.2	48.5	3.9	5.6	50.1	1.8	1.4	20.9	1.7	1.4
Caviar2	5.6	8.6	70.3	4.7	8.5	63.1	45.6	62.3	65.4	2.2	1.8
Leno	17.8	6.5	13.2	9.2	11.9	7.8	12.7	9.9	17.2	5.5	5.9
Walking	11.4	1.9	3.3	2.9	9.9	2.2	22.0	1.8	3.7	2.0	2.2
DavidIndoor	148.7	3.1	34.3	49.4	13.4	10.8	6.3	3.5	13.4	3.2	3.8
Car4	179.8	2.9	60.1	12.3	18.8	16.4	3.3	4.3	37.2	3.0	4.6
Car11	63.9	2.1	43.5	27.1	25.1	1.7	4.1	2.0	1.8	2.2	2.2
Deer	92.1	127.5	66.5	11.9	25.7	38.4	69.8	8.0	9.2	8.5	13.9
Jumping	58.4	36.8	9.9	63.0	3.6	8.8	55.2	39.1	19.2	5.0	4.9
Face	48.8	69.7	134.7	141.4	22.3	148.9	16.5	95.1	127.2	24.1	12.5
Average	54.4	27.0	44.2	28.9	15.1	30.1	25.1	20.2	28.2	5.5	5.3
Speed(fps)	4	32	32	4	18	10	2	9	1	5	20

Table 2. Average overlap rates of tracking algorithms. The best three results are shown in red, blue and green fonts.

	FragT [1]	IVT [19]	MIL [2]	VTD [14]	TLD [12]	APGL1 [3]	LSAT [17]	ASLAS [11]	MTT [28]	OSPT [24]	Ours
Occlusion1	0.90	0.85	0.59	0.77	0.65	0.87	0.90	0.83	0.79	0.91	0.88
Occlusion2	0.60	0.59	0.61	0.59	0.49	0.70	0.33	0.81	0.72	0.84	0.83
Caviar1	0.68	0.28	0.25	0.83	0.70	0.28	0.85	0.90	0.45	0.89	0.89
Caviar2	0.56	0.45	0.26	0.67	0.66	0.32	0.28	0.35	0.33	0.71	0.79
Leno	0.72	0.86	0.78	0.75	0.73	0.82	0.76	0.81	0.70	0.87	0.87
Walking	0.52	0.73	0.55	0.69	0.55	0.64	0.36	0.74	0.75	0.73	0.74
DavidIndoor	0.09	0.69	0.23	0.23	0.50	0.63	0.72	0.77	0.53	0.76	0.76
Car4	0.22	0.92	0.34	0.73	0.64	0.70	0.91	0.89	0.53	0.92	0.83
Car11	0.09	0.81	0.17	0.43	0.38	0.83	0.49	0.81	0.58	0.81	0.80
Deer	0.08	0.22	0.21	0.58	0.41	0.45	0.35	0.62	0.60	0.61	0.56
Jumping	0.14	0.28	0.53	0.08	0.69	0.59	0.09	0.24	0.30	0.69	0.68
Face	0.39	0.44	0.15	0.24	0.62	0.14	0.69	0.21	0.26	0.68	0.75
Average	0.42	0.59	0.39	0.55	0.59	0.58	0.56	0.67	0.55	0.79	0.78
Speed(fps)	4	32	32	4	18	10	2	9	1	5	20

cur simultaneously. The IVT [19] method is sensitive to partial occlusion since the Gaussian noise assumption cannot model outliers. The MIL [2] method does not perform well when the tracked object is occluded by a similar object (e.g., *Caviar1* and *Caviar2*) since the Haar-like features they used are less effective to distinguish similar objects when they are occluded by each other. Although the APGL1 [3] tracker explicitly considers partial occlusion by using a set of trivial templates, it also performs not well in some cases (e.g., *Caviar1* and *Caviar2*) as the raw pixel templates cannot always capture stable visual information.

Figure 4 (e)-(l) show representative results on eight image sequences which highlight other challenging factors (e.g., pose change, illumination variation, background clutter, fast motion). Our tracker also performs well in these cases, which can be attributed to two main reasons. First, the appearance change of the object in these cases can be well approximated by a PCA subspace [19]. Second, the estimated inlier-outlier mask is able to indicate unexpected factors, including the local illumination variation (e.g., *Car4* #0185 and *Car4*#0280), the background pixels within the object region (e.g., *DavidIndoor*#0325, *DavidIndoor*#0365 and *Deer*#0025), the blurred object region (e.g., *Jumping*#0035) and so on. Thus, the inlier-outlier mask makes that our tracker is able to focus on more stable object regions and achieve good performance.

Quantitative evaluation: To assess the performance of the proposed tracking algorithm, we adopt two popular criteria, the center location error and the overlap rate, in this paper. Table 1 shows the average center location error between our tracker and other competing algorithms, in which a small error value means a more accurate result. Table 2 reports quantitative comparisons between our tracker and other competing algorithms, in which the PASCAL overlap rate criterion [11, 24] is adopted to measure the accuracy of a tracker (a large overlap score means a more accurate result). We can see from these tables that our tracker achieves very favorable performance in terms of both accuracy and speed. Although the OSPT [24] method also achieves similar performance in terms of accuracy, it runs much slower than our tracker.

The effects of critical parameters: We note that the regularization parameters λ and λ_{ij} are two critical parameters in the proposed PCOM method. Figure 5 illustrates the average overlap scores of the proposed tracker with varied λ and λ_{ij} values. The parameter λ controls the level of outliers. If λ is too small, some inliers are identified as outliers. Hence, the solution of our model is not stable. If λ is too large, some outliers are treated as inliers and therefore lead to an incorrect solution. In addition, the parameter λ_{ij} also should be moderate to avoid over-sparsity or over-smoothness for the inlier-outlier mask. In this study, we choose $\lambda = 0.08$ and $\lambda_{ij} = 0.02$ as the default parameters for our PCOM problem.

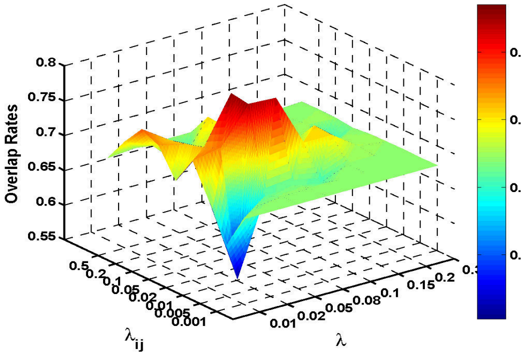


Figure 5. The effects of regularization parameters.

6. Conclusion

This paper presents a novel effective and fast tracking algorithm based on the proposed probability continuous outlier model (PCOM). In our PCOM method, the element of the noisy observation sample can be either represented by a PCA subspace with small Guassian noise or treated as an arbitrary value with a uniform prior, in which the spatial consistency prior is exploited. Then we derive the objective function of the PCOM method and present an iteration algorithm to solve it. The iteration process includes two basic steps: the outlier-free least squares regression and the standard max-flow/min-cut algorithm. Finally, we develop a generative tracker based on our PCOM method and a simple update scheme. Both qualitative and quantitative evaluations show that the proposed tracker achieves accurate and fast tracking performance. In the future, we will extend the proposed representation model for solving other vision problems (e.g., object recognition and motion estimation). In addition, we plan to integrate multiple visual cues (e.g., color and depth) into our method for more effective object tracking in different scenarios.

Acknowledgements: This work was supported by the Fundamental Research Funds for the Central Universities under Grant DUT13RC(3)105, and the Natural Science Foundation of China (NSFC) under Grants 61071209 and 61272372. This work was also supported by joint Foundation of China Education Ministry and China Mobile Communication Corporation under Grand MCM20122071.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, pages 798–805, 2006.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990, 2009.

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust ℓ_1 tracker using accelerated proximal gradient approach. In *CVPR*, pages 1830–1837, 2012.

[4] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, pages 105–112, 2001.

[5] CAVIAR. http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[6] D. Comaniciu, V. R. Member, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

[7] J. Fan, X. Shen, and Y. Wu. Scribble tracker: A matting-based approach for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1633–1644, 2012.

[8] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[9] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR*, pages 260–267, 2006.

[10] W. Hu, X. Li, W. Luo, X. Zhang, S. J. Maybank, and Z. Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2420–2440, 2012.

[11] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[14] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.

[15] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2009.

[16] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in Low Frame Rate Video: A cascade particle filter with discriminative observers of different life spans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1728–1740, 2008.

[17] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sparse appearance model and K-selection. In *CVPR*, pages 1313–1320, 2011.

[18] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. In *ICCV*, pages 1436–1443, 2009.

[19] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[20] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *IEEE International Conference on Computer Vision Workshops*, 2009.

[21] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *ICCV*, pages 1–8, 2007.

[22] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 2001.

[23] D. Wang, H. Lu, and M.-H. Yang. Least soft-threshold squares tracking. In *CVPR*, pages 2371–2378, 2013.

[24] D. Wang, H. Lu, and M.-H. Yang. Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing*, 22(1):314–325, 2013.

[25] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[26] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):1–45, 2006.

[27] K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *ECCV*, pages 864–877, 2012.

[28] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049, 2012.