PageRank Tracker: From Ranking to Tracking

Chen Gong, Keren Fu, Artur Loza, Qiang Wu, *Member, IEEE,* Jia Liu, and Jie Yang

***Abstract*—Video object tracking is widely used in many real-world applications, and it has been extensively studied for over two decades. However, tracking robustness is still an issue in most existing methods, due to the difficulties with adaptation to environmental or target changes. In order to improve adaptability, this paper formulates the tracking process as a ranking problem, and the PageRank algorithm, which is a well-known webpage ranking algorithm used by Google, is applied. Labeled and unlabeled samples in tracking application are analogous to query webpages and the webpages to be ranked, respectively. Therefore, determining the target is equivalent to finding the unlabeled sample that is the most associated with existing labeled set. We modify the conventional PageRank algorithm in three aspects for tracking application, including graph construction, PageRank vector acquisition and target filtering. Our simulations with the use of various challenging public-domain video sequences reveal that the proposed PageRank tracker outperforms mean-shift tracker, co-tracker, semiboosting and beyond semiboosting trackers in terms of accuracy, robustness and stability.**

***Index Terms*—PageRank, power method, robust tracking.**

I. Introduction

OBJECT tracking is one of the crucial fields of computer vision. It serves as the foundation of many areas such as intelligent surveillance, scene understanding and behavior analysis, etc. The goal of object tracking is to precisely associate the target in consecutive video frames while maintaining its identity. Although video tracking has been and still remains a subject of extensive research, the issue of robust tracking is still far from being resolved.

Conventional methods such as optical flow [1], [2], mean-shift [3], [4], and particle filter [5]–[7] are sensitive to the changes from target and the environment. Over time it has become clear that adaptive and stable tracking is difficult to achieve by simple matching or prediction mechanisms as in the above methods. In recent years, alternative approach that formulates tracking as a classification problem has gained great popularity, and has shown improved robustness for visual tracking. The representative classification-based methods can be grouped into three main types.

Manuscript received August 9, 2012; revised June 3, 2013; accepted July 13, 2013. Date of publication August 15, 2013; date of current version May 13, 2014. This work was supported by the National Science Foundation, China, under Grant 61273258. This paper was recommended by Associate Editor Q. Ji.

C. Gong, K. Fu, A. Loza, J. Liu and J. Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: goodgongchen@sjtu.edu.cn; fkrsuper@sjtu.edu.cn; artur_loza@sjtu.edu.cn; liujia1982@sjtu.edu.cn; jieyang@sjtu.edu.cn).

Qiang Wu is with the School of Computing and Communications, University of Technology, Sydney, NSW 2007, Australia (e-mail: qiang.wu@uts.edu.au).

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.2013.2274516

1) *Multiple instance learning (MIL):* This learning strategy regards training samples as bags containing multiple instances, and a bag is considered positive as long as it has at least one positive sample. Babenko *et al.* [8]–[10] firstly introduced this learning approach to visual tracking. Their approach was extended to weighted MIL (WMIL) tracker [11] by integrating the sample importance into an efficient online learning procedure. Ni *et al.* [12] combined the conventional particle filter tracker with MIL in order to explicitly handle the false positive samples.
2) *Semi-supervised learning (SSL):* The advantage of SSL is that it can deal with the situations where labeled samples are extremely scarce, as this learning approach also utilizes massive unlabeled samples to improve the classification accuracy. Co-training, graph regularization are representatives of this learning method [13]. Fang *et al.* [14] applied the co-training framework to tracking, which was extended in [15] by incorporating the third classifier for ambiguous situations. Harmonic functions [16] was adapted for tracking [17] to achieve robust and stable performance. Gong *et al.* [18] adopted linear neighborhood propagation (LNP) [19] and achieved encouraging results. In [20]–[22] online semi-supervised boosting was used in order to perform feature selection task, which leads to an improved performance. [23] exploited the structure of unlabeled data and proposed the P-N learning algorithm consisting of positive and negative constraints. P-N learning was adapted in tracking-learning-detection approach, which was shown to be effective for face tracking [24]. Other works belonging to this type include [25]–[27], etc.
3) *Combination of trackers:* Algorithms with this type usually contain several, often interactive, trackers. Different trackers are fused or selected during tracking process so that their respective strength and complementarity can be fully utilized. Zhong *et al.* [28] developed an online strategy for evaluating the performances of different trackers in order to solve the chicken-and-egg problem, and thus guaranteeing all trackers are updated reliably. Kwon *et al.* [29] proposed a visual tracker sampler, which dynamically samples several good trackers from the tracker space.

Above classification-based trackers have shown promising performance. However, they mainly have two shortcomings. Firstly, although the goal of classification-based methods is to distinguish the target region from complicated background, the environment varies broadly during the tracking process. As a consequence, the background cannot be sufficiently described by limited negative samples, which may cause erroneous classification. Secondly, the quantity of positive and negative samples is imbalanced (the negative samples outnumber significantly the positive samples), which will probably influence the tracking results. Nevertheless, though the above three types of classification-based methods differ from each other and have some defects, all of them contain a learning procedure, that is, the trackers are updated continually so that the variations of both target and environment can be learned timely, which is a critical step toward robust tracking.

This paper regards tracking as a ranking problem, and a novel PageRank tracker is proposed, in which robust tracking is achieved by continually learning the graph in the PageRank method. To the best of our knowledge, PageRank algorithm has been already successfully applied to webpage searching [30], multimedia retrieval [31] and document retrieval [32], but it has not been used for visual tracking. For detailed introduction of PageRank method, the reader is referred to [30] and [33].

Applying ranking methodology to tracking has three advantages. Firstly, the goal of tracking is indeed to find one unlabeled sample that represents the known target most properly, which is also the main task of the PageRank technique. Secondly, the samples collected in tracking process are not independent and they are subjected to spatial and temporal constraints [23]. The relationship between the samples can be perfectly described by the graph in the PageRank tracker. Thirdly, the sample imbalance does not affect the PageRank algorithm because the goal of PageRank is to determine the most relevant sample given the known target. The main contributions of this paper are summarized below.

1) A suitable application of the PageRank framework to visual tracking problem.
2) A novel method for graph construction by considering both neighborhood and pairwise information among samples.
3) A modified iterative expression for solving the PageRank vector, that includes historical records of target.
4) A new way for filtering out the false positive outputs.

The remainder of this paper is organized as follows. In Section II the motivation and framework of the proposed PageRank tracker are explained. Section III introduces the algorithm for obtaining transition probability matrix. Power method for solving the PageRank vector is presented in Section IV. In Sections V and VI, the methods for target filtering and labeled set updating are presented, respectively. Empirical studies of the proposed tracker are subject of Section VII. Finally, a conclusion is drawn in Section VIII.

II. Overview of the Proposed Method

This section briefly explains how to adapt the traditional PageRank algorithm to the new tracking domain, and then we define some useful notations and introduce the complete tracking framework.

A. From Ranking to Tracking

In the traditional graph-based model, the graph is built as $G = \langle V, E\rangle$ where V is vertex set and E is edge set. The goal of PageRank algorithm is to rank webpages according to their relevances with certain queries. Consequently, webpages are represented by vertices, and the hyperlinks connecting them are expressed by edges. When it comes to object tracking, the target regions in previous frames (labeled samples) are analogous to the query webpages, and the sampled regions in current frame (unlabeled samples) are analogous to the potential webpages to be ranked. The task of finding the target is thus equivalent to finding an unlabeled sample that is most related to the existing labeled instances. The motivation of the proposed method is that selecting the target from numerous unlabeled samples can be formulated in a similar way as ranking webpages. However, different from ranking webpages, the "hyperlinks" between samples no longer exist in tracking application, and thus edges for tracking should be redefined in new ways. Based on these considerations, in our case V is the set of labeled and unlabeled samples, and E is the set of constraints between these samples. More detailed explanations of this formulation can be found in Section III.

B. PageRank Tracker Framework

According to the core idea explained in Section II-A, a PageRank tracker is designed as follows. Firstly, the user manually specifies a target TG in the first frame, then in the following $S-1$ frames (S should not be set very big) a simple tracker, for example, the mean-shift tracker [3], is utilized to collect a small amount of positive samples (i.e. the target tracked by mean-shift in every frame) to establish labeled dataset $LS = \{L_i | i = 1, 2, \cdots, l\}$. Following this, the mean-shift tracker is replaced by PageRank tracker. For each new frame, a large number of unlabeled samples are collected around the location of the target in previous frame, which form the unlabeled set $US = \{U_i | i = 1, 2, \cdots, u\}$. Then based on the union of US and LS, a graph G is established with its adjacency matrix named as transition probability matrix (denoted as $\mathbf{P}$). A power method is used to solve the PageRank vector $\mathbf{f}^*$, according to which the top-ranked sample $\mathbf{x}_{top}$ is generated. After that, a score of $\mathbf{x}_{top}$ is calculated by integrating the recommendation level of ranking process and its similarity with LS, and $\mathbf{x}_{top}$ is taken as the final target TG if the score is larger than an adaptive threshold. Finally, the sample corresponding to the target region is added to LS for graph updating in the next T frames. This whole procedure is presented in Fig. 1.

III. Finding Transition Probability Matrix

In the conventional PageRank algorithm, the transition probability matrix $\mathbf{P}_{n\times n}$ (where n is the amount of webpages in the whole graph) of G is specified by webpages and hyperlinks. The element p_{ij} in $\mathbf{P}_{n\times n}$ stands for the probability of switching from webpage i to webpage j, which satisfies $p_{ii} = 0$ $(1 \le i \le n)$ and $\sum_{j=1}^{n} p_{ij} = 1$. In the proposed tracking system, the samples previously regarded as the target are labeled as positive, and the samples collected during tracking process are considered as unlabeled, so p_{ij} reflects how closely the sample $\mathbf{x}_i$ is linked to $\mathbf{x}_j$. In other words, p_{ij} is the edge weight $\mathbf{x}_i$ to $\mathbf{x}_j$ in graph G. If the whole dataset $DS = LS \cup US$

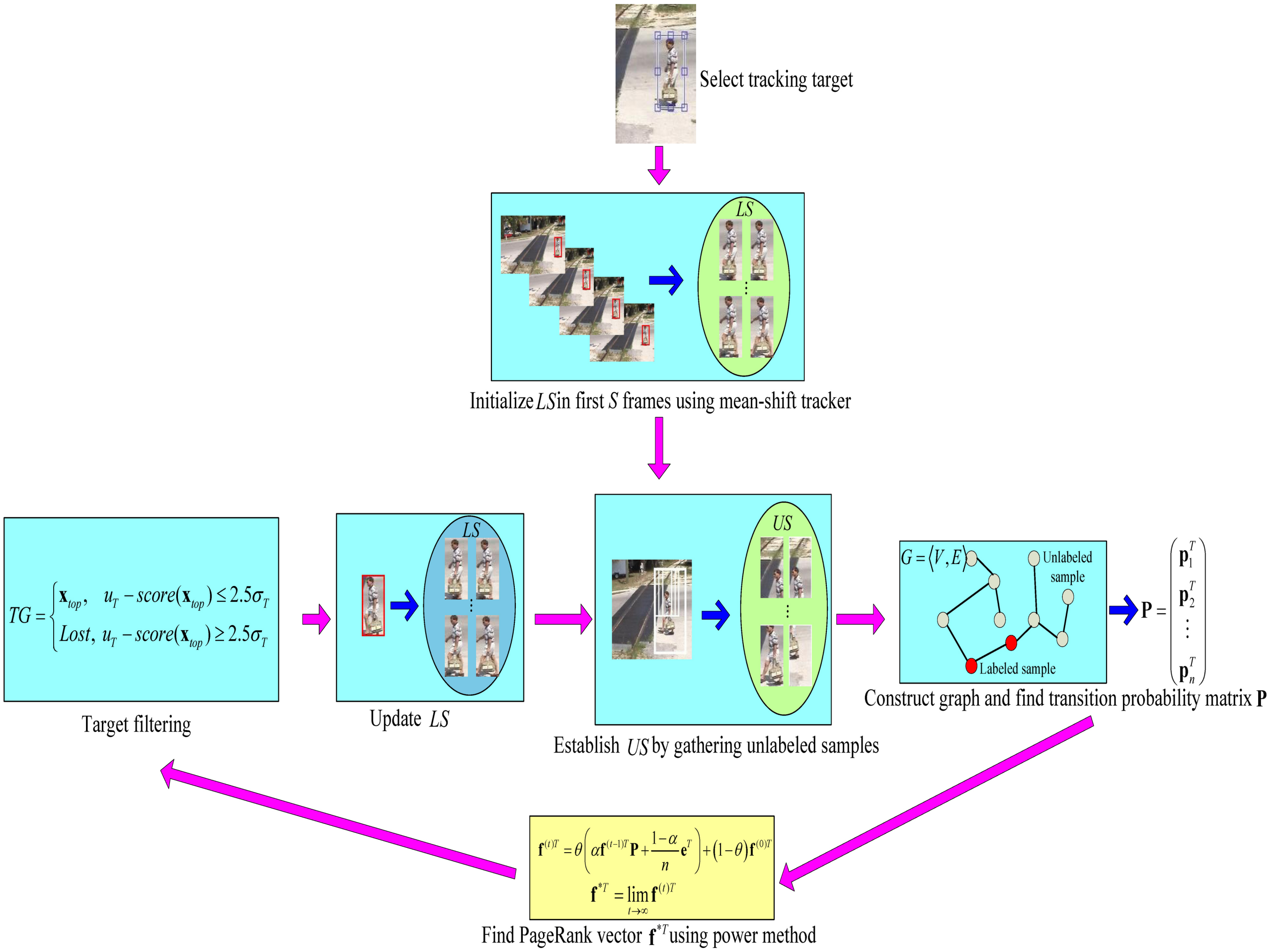


Fig. 1. Framework of PageRank tracker.

is denoted by $DS = \{\mathbf{x}_i | i = 1, 2, \cdots, n, n = l + u\}$, then it is possible to use Gaussian kernel function [16] to describe such an edge weight

$$p_{ij} = exp(-\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2\sigma^2}). \tag{1}$$

Equation (1) reflects the pairwise relationship between different samples but ignores the neighborhood information of them. Moreover, the parameter of kernel width σ is also difficult to choose. An alternative approach for creating graph was proposed in [19], which assumed that each data point in the graph can be optimally reconstructed using a linear combination of its neighbors. Consequently, solving the edge weights between samples is equivalent to finding the solutions of the following optimization problem

$$\begin{array}{ll} \min_{p_{ij}} & \epsilon_i = \| \mathbf{x}_i - \sum_{j:\mathbf{x}_{ij} \in N(\mathbf{x}_i)} p_{ij}\mathbf{x}_{ij} \|^2 \\ s.t. & \sum_j p_{ij} = 1, \quad p_{ij} \geq 0 \end{array} \tag{2}$$

where $N(\mathbf{x}_i)$ denotes the neighbors of $\mathbf{x}_i$, and p_{ij} is the contribution of $\mathbf{x}_{ij}$ to $\mathbf{x}_i$. This method utilises the neighborhood information of each data point, while ignores the pairwise relationship.

In order to fuse pairwise information in the graph construction process, (1) is modified as follows. Suppose $\mathbf{x}_{ij}$, $\mathbf{x}_{ik}$ are two of $\mathbf{x}_i$'s K neighbors, and p_{ij}, p_{ik} are edge weights accordingly, then the more similar one sample is to $\mathbf{x}_i$, the larger edge weight should be assigned to, namely

$$\frac{\| \mathbf{x}_i - \mathbf{x}_{ij} \|}{\| \mathbf{x}_i - \mathbf{x}_{ik} \|} = \frac{p_{ik}}{p_{ij}}. \tag{3}$$

$\| \cdot \|$ in (3) denotes $\mathcal{L}_2$-norm. Note that by using this form we avoid introducing σ by taking $\mathbf{x}_i$ as a bridge. In fact, (3) can be rewritten in an optimization expression

$$\min \, (p_{ij} \| \mathbf{x}_i - \mathbf{x}_{ij} \| - p_{ik} \| \mathbf{x}_i - \mathbf{x}_{ik} \|)^2. \tag{4}$$

Given all these factors, (2) and (4) are integrated into one expression by summing over all pairs of neighbors of $\mathbf{x}_i$

$$\begin{aligned} &\min_{p_{ij}} \epsilon_i = \| \mathbf{x}_i - \sum_{j=1}^{K} p_{ij}\mathbf{x}_{ij} \|^2 + \\ &\gamma \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} (p_{ij} \| \mathbf{x}_i - \mathbf{x}_{ij} \| - p_{ik} \| \mathbf{x}_i - \mathbf{x}_{ik} \|)^2 \\ &s.t. \quad \sum_j p_{ij} = 1, \quad p_{ij} \geq 0. \end{aligned} \tag{5}$$

The first term of the objective function in (5) is equivalent to (2), which reflects the local property of $\mathbf{x}_i$. The second term is the summation form of (4), which describes the pairwise relationship in the neighborhood of $\mathbf{x}_i$. The free parameter γ

balances the weight of these two terms, and it is usually set to ten. For simplicity, (5) is formulated in a matrix form

$$\begin{array}{ll}\min_{\mathbf{p}_i} & \epsilon(\mathbf{p}_i) = \frac{1}{2}\mathbf{p}_i^T\mathbf{H}\mathbf{p}_i + \mathbf{d}^T\mathbf{p}_i + q \\ s.t. & \mathbf{A}\mathbf{p}_i = \mathbf{b}, \quad \mathbf{p}_i \geq 0\end{array} \tag{6}$$

where

$$\mathbf{p}_i = \begin{pmatrix} p_{i1} & p_{i2} & \cdots & p_{iK}\end{pmatrix}^T$$

and $\mathbf{H}$ is a $K \times K$ matrix defined by

$$\mathbf{H}_{jk} = \begin{cases} 2(\| \mathbf{x}_{ij} \|^2 + \gamma(K-1) \| \mathbf{x}_i - \mathbf{x}_{ij} \|^2) & j = k \\ 2(\mathbf{x}_{ij}^T\mathbf{x}_{ik} - \gamma \| \mathbf{x}_i - \mathbf{x}_{ij} \| \| \mathbf{x}_i - \mathbf{x}_{ik} \|) & j \neq k \end{cases}$$

$$\mathbf{d} = \begin{pmatrix} -2\mathbf{x}_i^T\mathbf{x}_{i1} & \cdots & -2\mathbf{x}_i^T\mathbf{x}_{ik} & \cdots & -2\mathbf{x}_i^T\mathbf{x}_{iK}\end{pmatrix}^T$$

$$q = \| \mathbf{x}_i \|^2 \quad \mathbf{A} = \underbrace{\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}}_{K} \quad \mathbf{b} = \underbrace{\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}}_{K}^T.$$

By solving n standard quadratic programming problems as (6), vectors $\mathbf{p}_1$ to $\mathbf{p}_n$ are calculated and the adjacency matrix $\mathbf{P}$ is formulated as

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_n^T \end{pmatrix}. \tag{7}$$

However, $\mathbf{P}$ is usually a reducible sparse matrix, and the stationary vector solved by the following iterative process may not exist, so a revision is needed to make $\mathbf{P}$ irreducible

$$\overline{\mathbf{P}} = \alpha\mathbf{P} + \frac{1-\alpha}{n}\mathbf{E} \tag{8}$$

where $\mathbf{E}$ is an all-1 matrix that has the same size as $\mathbf{P}$. α is in the range (0, 1) and is usually set to 0.85 in order to achieve the fastest convergence rate [33]. $\overline{\mathbf{P}}$ is an irreducible and stochastic matrix with all elements above zero, which constitutes the transition probability matrix.

IV. Finding PageRank Vector

Based on the transition probability matrix $\overline{\mathbf{P}}$, this section aims to calculate the stationary PageRank vector, according to which the unlabeled samples are ranked. The basic solution presented in Section IV-A is computational expensive, so an improved power method is introduced in Section IV-B to accelerate the processing speed.

A. Basic Solution

One common way to find the PageRank vector is to set the initial vector $\pi^{(0)T} = \frac{1}{n}\mathbf{e}^T$ with $e_i = 1$ $(1 \leq i \leq n)$, and then the PageRank vector can be calculated through an iteration

$$\pi^{(t)T} = \pi^{(t-1)T}\overline{\mathbf{P}}. \tag{9}$$

The irreducible property of $\overline{\mathbf{P}}$ ensures that this iteration converges to a stationary PageRank vector π^*, in which every element represents the recommendation level of corresponding webpage. It is widely known that this process is a Markov chain, namely π^* is independent of the initial state $\pi^{(0)}$ [34].

However, there are differences between webpage ranking and object tracking in terms of the initial state $\pi^{(0)}$. The final ranking ordering of webpages should not be influenced by the $\pi^{(0)}$ that indicates their initial importance for ranking, because whether a webpage should be recommended or not, depends only on how closely it relates to the current query webpage. In contrast to webpage ranking, the location of target in tracking depends on the history of target in previous frames, so the final ranking results should not be irrelevant to $\pi^{(0)}$. In our tracking system, elements in $\pi^{(0)}$ are defined as

$$\pi_i^{(0)} = \begin{cases} \frac{1}{l}, & \mathbf{x}_i = \text{labeled positive} \\ 0, & \mathbf{x}_i = \text{unlabeled} \end{cases} \tag{10}$$

where

$$\sum_{i=1}^{n} \pi_i^{(0)} = 1.$$

Therefore, the historical information of the target (labeled positive samples) is recorded into $\pi^{(0)}$, which plays an important role in finding the PageRank vector. In order to include this historical information, (9) is modified so that the iteration is no longer a Markov process

$$\mathbf{f}^{(t)T} = \theta\mathbf{f}^{(t-1)T}\overline{\mathbf{P}} + (1-\theta)\,\mathbf{f}^{(0)T} \tag{11}$$

where the notation $\mathbf{f}$ is equivalent to π in (9) and $\theta \in (0, 1)$. (11) reveals that $\mathbf{f}^{(t)T}$ can be regarded as the convex combination of initial state $\mathbf{f}^{(0)T}$ and the expression simply representing Markovian idea. It is proved (see Appendix) that after iterations, (11) finally converges to

$$\mathbf{f}^{*T} = \lim_{t\to\infty}\mathbf{f}^{(t)T} = (1-\theta)\,\mathbf{f}^{(0)T}\left(\mathbf{I} - \theta\overline{\mathbf{P}}\right)^{-1} \tag{12}$$

which suggests that the PageRank vector $\mathbf{f}^{*T}$ not only relates to transition probability matrix $\overline{\mathbf{P}}$, but also relies on the initial information $\mathbf{f}^{(0)T}$ as we expected. In the proposed tracking algorithm, convergence criterion is that the ordering of elements' values in $\mathbf{f}^{(t)T}$ does not change for certain successive iterations, and this measure reduces the iteration times significantly [35].

B. Power Method

In practice, using (11) directly for iteration is inefficient because in every iteration the power of $\overline{\mathbf{P}}$, which is a large, dense matrix, has to be calculated and stored. In order to address this problem, we substitute (8) into (11), and obtain

$$\begin{aligned}\mathbf{f}^{(t)T} &= \theta\mathbf{f}^{(t-1)T}\overline{\mathbf{P}} + (1-\theta)\mathbf{f}^{(0)T} \\ &= \theta\mathbf{f}^{(t-1)T}\left(\alpha\mathbf{P} + \frac{1-\alpha}{n}\mathbf{E}\right) + (1-\theta)\,\mathbf{f}^{(0)T} \\ &= \theta\mathbf{f}^{(t-1)T}\left(\alpha\mathbf{P} + \frac{1-\alpha}{n}\mathbf{e}\mathbf{e}^T\right) + (1-\theta)\,\mathbf{f}^{(0)T} \\ &= \theta\left(\alpha\mathbf{f}^{(t-1)T}\mathbf{P} + \frac{1-\alpha}{n}\mathbf{e}^T\right) + (1-\theta)\,\mathbf{f}^{(0)T}.\end{aligned} \tag{13}$$

In the above derivation, two basic properties are taken into account, i.e. $\mathbf{E} = \mathbf{e}\mathbf{e}^T$ and $\mathbf{f}^{(t-1)T}\mathbf{e} = 1$. By using (13) the computation of power of matrix is avoided. Moreover, since $\mathbf{P}$ is a sparse matrix, (13) reduces computational complexity as well as storage requirements. Above process is called the power method.

Alternatively, $\mathbf{f}^{*T}$ can also be calculated directly by solving linear equations. Suppose the stationary vector $\mathbf{f}^{*T}$ is reached, then the following equation holds

$$\mathbf{f}^{*T} = \theta\left(\alpha\mathbf{f}^{*T}\mathbf{P} + \frac{1-\alpha}{n}\mathbf{e}^T\right) + (1-\theta)\,\mathbf{f}^{(0)T} \tag{14}$$

i.e.,

$$\mathbf{f}^{*T}\left(\mathbf{I} - \theta\alpha\mathbf{P}\right) = \theta\frac{1-\alpha}{n}\mathbf{e}^T + (1-\theta)\,\mathbf{f}^{(0)T}. \tag{15}$$

Equation (15) is a nonhomogeneous system of linear equations that can be used to solve $\mathbf{f}^{*T}$. However, the complexity of this way is very high because the coefficient matrix $\mathbf{I}-\theta\alpha\mathbf{P}$ is usually very large. Therefore, we adopt the power method to work out $\mathbf{f}^{*T}$.

V. Locating the Target

Given $\mathbf{f}^* = \begin{pmatrix} f_1^* & f_2^* & \cdots & f_l^*, & f_{l+1}^* & f_{l+2}^* & \cdots & f_{l+u}^* \end{pmatrix} \triangleq \begin{pmatrix} F_l^*, & F_u^* \end{pmatrix}$, a simple and straightforward way is to take the sample that ranks first in F_u^* (denoted as $\mathbf{x}_{top}$) as the target TG. However, a number of experiments reveal that this top-ranked sample does not always precisely represent the target region when disturbances happen. Employing these false positive samples as target will impair the tracking performance significantly. Because the sample that represents target is labeled positive for future training, so the erroneous updating will probably lead to drifting problem. Therefore, a filter is needed to exclude such inaccurate target location.

Whether $\mathbf{x}_{top}$ can be the valid target depends on two factors: one is the recommendation level of ranking algorithm, the other is the similarity between $\mathbf{x}_{top}$ and the previous appearances of target. The elements f_i^* $(i = l+1, l+2, \cdots, l+u)$ in F_u^* can be regarded as the recommendation scores of $\mathbf{x}_i$ to be the target. Therefore, to what degree the top-ranked sample $\mathbf{x}_{top}$ can be selected as the final target is revealed by f_{top}^*.

Moreover, in tracking situation the object should obey the temporal constraint [23] that its appearance does not vary significantly during a short time, so if $\mathbf{x}_{top}$ is very similar to the elements in labeled set LS, then it is very likely to be the target. Based on this consideration, the similarity between $\mathbf{x}_{top}$ and LS is defined as

$$sim\left(\mathbf{x}_{top}, LS\right) = \sum_{i=1}^{l} \eta^i \arccos\left(\mathbf{x}_{top}, L_i\right) \tag{16}$$

in which L_i $(1 \le i \le l)$ are the i-th latest records in LS. $\arccos\left(\mathbf{x}_{top}, L_i\right)$ represents the similarity between $\mathbf{x}_{top}$ and L_i by calculating the arccosine value of their angles. $\eta \in (0, 1)$ is the time dampening factor so that the similarity between $\mathbf{x}_{top}$ and the later L_i gains larger weight.

Therefore, a score is defined to evaluate the confidence level of $\mathbf{x}_{top}$ to be the target, which is formulated as

$$score\left(\mathbf{x}_{top}\right) = f_{top}^* sim\left(\mathbf{x}_{top}, LS\right). \tag{17}$$

Low $score\left(\mathbf{x}_{top}\right)$ usually means that PageRank tracker cannot locate the object accurately. This usually happens when target is changing appearance or occluded by other objects significantly. Therefore, similar to that in [18] and [36], an adaptive threshold is designed to filter out the inaccurate locations. Suppose $score(\mathbf{x}_{top})$ in the j'th frame is denoted as $score_j(\mathbf{x}_{top})$, then the mean value and standard deviation of $score_j(\mathbf{x}_{top})$ in previous T frames are

$$u_T = \frac{1}{T}\sum_{j=1}^{T} score_j(\mathbf{x}_{top}) \tag{18}$$

$$\sigma_T = \sqrt{\frac{1}{T}\sum_{j=1}^{T}\left[score_j(\mathbf{x}_{top}) - u_T\right]^2} \tag{19}$$

respectively. Here we assume that the scores of $\mathbf{x}_{top}$ in the latest T frames obey the Gaussian distribution. Therefore, a sudden drop of $score(\mathbf{x}_{top})$ that is $2.5\sigma_T$ away from u_T indicates the target is lost, and thus the TG in current frame is determined by

$$TG = \begin{cases} \mathbf{x}_{top}, & u_T - score(\mathbf{x}_{top}) \le 2.5\sigma_T \\ \text{Lost}, & u_T - score(\mathbf{x}_{top}) \ge 2.5\sigma_T. \end{cases} \tag{20}$$

VI. Updating Labeled Set

The samples corresponding to accurate tracking are labeled as positive and added to LS for graph construction in coming frames. Although the approach explained in Section V is very simple, it ensures that graph updating in every frame is accurate, which is very important for preventing drifting in self-learning framework like ours. Moreover, out-of-date samples before T frames are trimmed off from LS in order to let PageRank tracker learn the latest appearance of object in time.

VII. Experimental Results

In this section, we firstly demonstrate that the tracking results can be substantially improved due to the proposed modifications on the original PageRank algorithm. Then the performance of the proposed PageRank tracker (abbreviated as PRT) is compared with other four popular trackers, including mean-shift tracker [3], co-tracker [14] semi-boosting tracker [21] and beyond semi-boosting tracker [22] (abbreviated as MS, CT, SB, BSB, respectively), by tracking objects on some challenging public sequences. Finally, the reasons that our PageRank tracker outperforms above baselines are briefly discussed. In all the experiments below, we set $K = 7$, $S = 5$, $\theta = 0.05$, $\eta = 0.9$ and $C = 5$ in PageRank tracker. T is also tuned properly in all the videos in order to achieve best performance. Gabor feature [37] is adopted and all feature vectors are reduced to 35 dimensions using principle component analysis (PCA) to characterize the related samples.

A. Validation of PageRank Tracker

In this paper, three major modifications on the conventional PageRank method were performed in order to make it applicable to object tracking.

(i) *Graph construction:* A novel graph construction algorithm is proposed by adding a pairwise term to the objective function of (2).

TABLE I
FOUR SIMULATIONS AND THEIR ABBREVIATIONS

Simulation	Graph construction	Iteration implementation	abbreviation
1	(2)	(9)	local + original
2	(2)	(13)	local + revised
3	(5)	(9)	local, pairwise + original
4	(5)	(13)	local, pairwise + revised

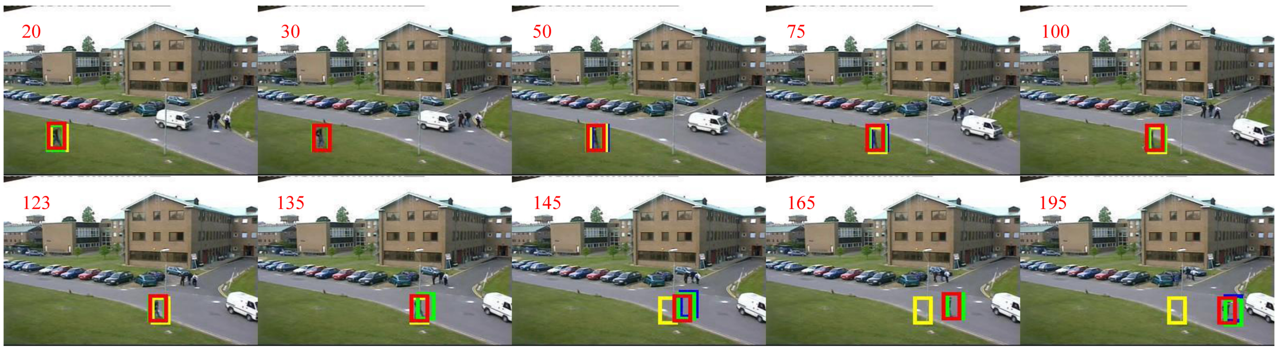


Fig. 2. Validation of the modifications (red: "local, pairwise + revised"; green: "local, pairwise + original"; blue: "local + original"; yellow: "local + revised").

(ii) *PageRank vector acquisition:* The original iterative expression (9) is replaced by (13) because we hold that the historical information of the target should be considered for robust tracking.

(iii) *Target filtering:* A target filtering scheme is proposed to exclude the possible false positive samples that rank first.

This section aims to demonstrate that the above three modifications are critical to accurate tracking through both qualitative and quantitative evaluations.

1) *Proof of (i):* In order to illustrate the effect of (i), we observe the tracking performances under four different simulations listed in Table I. Simulations 1 and 2 simply adopt the local information of samples for graph construction, which is described by (2). Comparatively, Simulations 3 and 4 add the pairwise information to graph establishment [see (5)] in order to boost tracking performance. For simplicity, the abbreviations of the four simulations are given in the last column of Table I. Note that the Simulation 4 is actually the proposed PageRank Tracker.

We firstly adopt the *Pedestrian1*[1] sequence to test the performances of the above four simulations, and their results are shown in Fig. 2. It can be observed that if "local + revised" is adopted, the tracker fails as soon as the man is occluded by the lamppost (yellow box). However, if the pairwise term is incorporated to construct the graph G (namely "local, pairwise + revised" denoted by red box), the revised PageRank algorithm is able to track the object precisely. Figure 3 plots the position errors of four simulations according to the manually annotated groundtruth. The effect of modification (i) is verified by comparing the red curve with the yellow one. We see that without the pairwise term, the tracker suffers from drifting problem after frame 145, which is also illustrated in Fig. 2. Therefore, the pairwise term added by us is critical to robust tracking. More interestingly, from blue and green curves we find that even we adopt the original iterative expression (9) to obtain the PageRank vector $\mathbf{f}^{*T}$, the performance can be slightly improved by incorporating the pairwise term. Generally, the proposed PageRank tracker ("local, pairwise + revised") represented by red curve obtains the minimum position error among the four simulations.

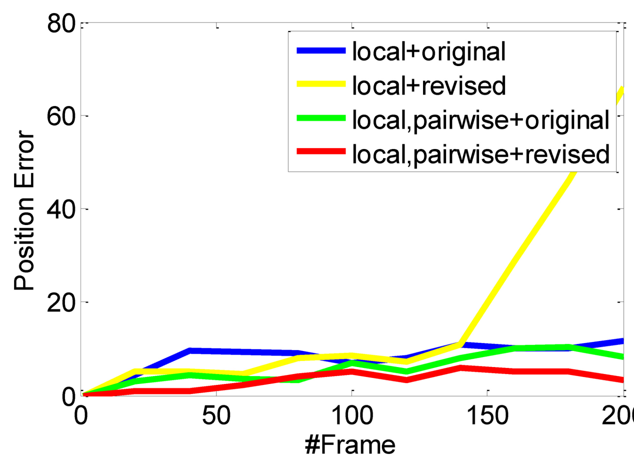


Fig. 3. Position error of the four different simulations on *Pedestrian1* sequence.

We also conduct the four simulations on *Pedestrian2*[2] sequence [Fig. 6(a) for example frames], and their position errors are plotted in Fig. 4. By comparing "local + original" vs. "local, pairwise + original" and "local + revised" vs. "local, pairwise + revised", we observe that the incorporation of pairwise term decreases the tracking error significantly no matter (9) or (13) is utilized for iteration.

[1]ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/

[2]http://www.cs.cmu.edu/ yaser/

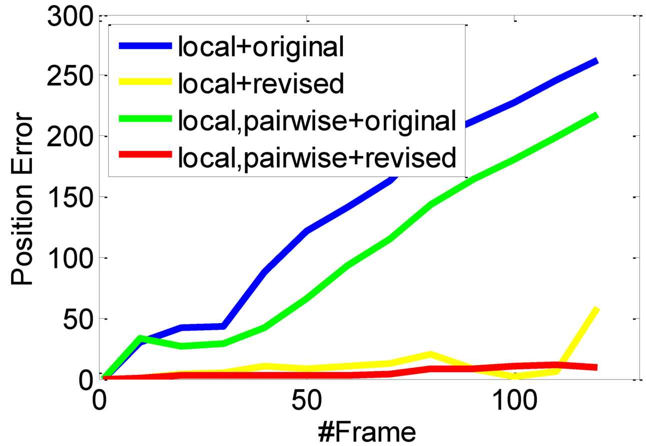


Fig. 4. Position error of the four different simulations on *Pedestrian2* sequence.

2) *Proof of (ii):* In Table I, simulation 3 simply uses the original iterative expression (9) of PageRank algorithm to obtain the PageRank vector. By contrast, Simulation 4 takes the revised form (13) for iterative procedure in order to achieve more encouraging results. Therefore, the improved results brought by modification (ii) can be observed by comparing the simulations “local, pairwise + original” with “local, pairwise + revised” in Figs. 2–4 (red and green boxes or curves). In Fig. 2, we see that when the pedestrian is walking from grassland to road (frames 135–195), the green tracking box begins to locate the target with slight inaccuracy. However, the performance is significantly enhanced by replacing the revised iterative expression (9) with (13). Figs. 3 and 4 also illustrate that the tracking trajectories of “local, pairwise + revised” are closer to groundtruth than other simulations on *Pedestrian1* and *Pedestrian2* sequences.

3) *Proof of (iii):* To demonstrate the effectiveness of modification (iii), we apply the proposed PageRank tracker to tracking the face appeared in *Dudek*[3] sequence. From Fig. 5 we see that the target is significantly occluded by the hand during frames 204–225, in which all the samples should not be taken as targets even though some of them are perhaps top-ranked by the ranking process. The occlusion will obviously decrease $score\left(\mathbf{x}_{top}\right)$, which helps the filtering scheme in Section V to exclude the samples recommended by the PageRank algorithm. The 2nd column of Fig. 5 illustrates the occlusion process, and the top-ranked samples during this process are presented in the 3rd column. The figure reveals that the samples that rank first are correctly filtered out by the modification (iii) when occlusion happens, which reflects that the proposed filtering methodology can detect the false positive samples effectively.

B. Resistance to Background Change

Background change is often the key reason for drifting. The proposed PageRank tracker (PRT) and another four popular trackers, MS, CT, SB, BSB, are tested on the sequence *Pedestrian2*. Fig. 6(a) shows some representative frames for

[3]http://www.cs.toronto.edu/vis/projects/dudekfaceSequence.html

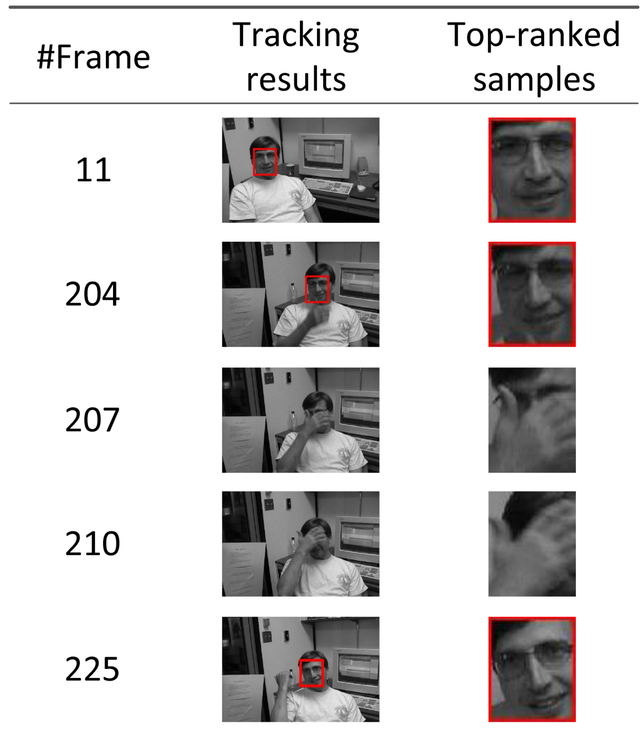


Fig. 5. Target filtering in the occlusion process. (The top-ranked samples taken as valid targets are surrounded by red rectangles in the third column. The samples that are filtered out by modification (iii) are not marked by red rectangles).

comparison. There are two difficulties in this sequence: one is the sudden background change when the pedestrian steps into or step out of the black region, the other is the view change of observing the target when a car is passing by. All the trackers are able to locate the target precisely at the beginning phase (frame 10). However, when the pedestrian walks through the black region (frame 23–50), MS (blue box) and CT (yellow box) fail immediately, and their tracking boxes are reluctant to enter into the black region. SB (cyan box) and BSB (magenta box) also fail to find the target, and thus their tracking boxes do not appear in these frames. However, BSB is able to redetect the pedestrian as shown in frames 73 and 84. During the frames 95–125 when the man begins to go across the road, the observing view changes and this leads to the tracking failure of BSB again. In contrast to MS, CT, SB and BSB, the proposed PRT is not influenced by the background or view changes, and it tracks the target successfully throughout the sequence. The curves of trajectory error in Fig. 7(a) also demonstrates this point.

C. Tracking Object Similar to Background

The trackers are easily confused if the target is very similar to the background. Fig. 6(b) shows the *Stone* sequence[4] in which we aim to track a yellow cobblestone located among many similar cobblestones. PRT and all the baselines are implemented to see whether they are able to distinguish the target from the background. Frame 78 reveals that when the

[4]http://ice.dlut.edu.cn/lu/publications.html

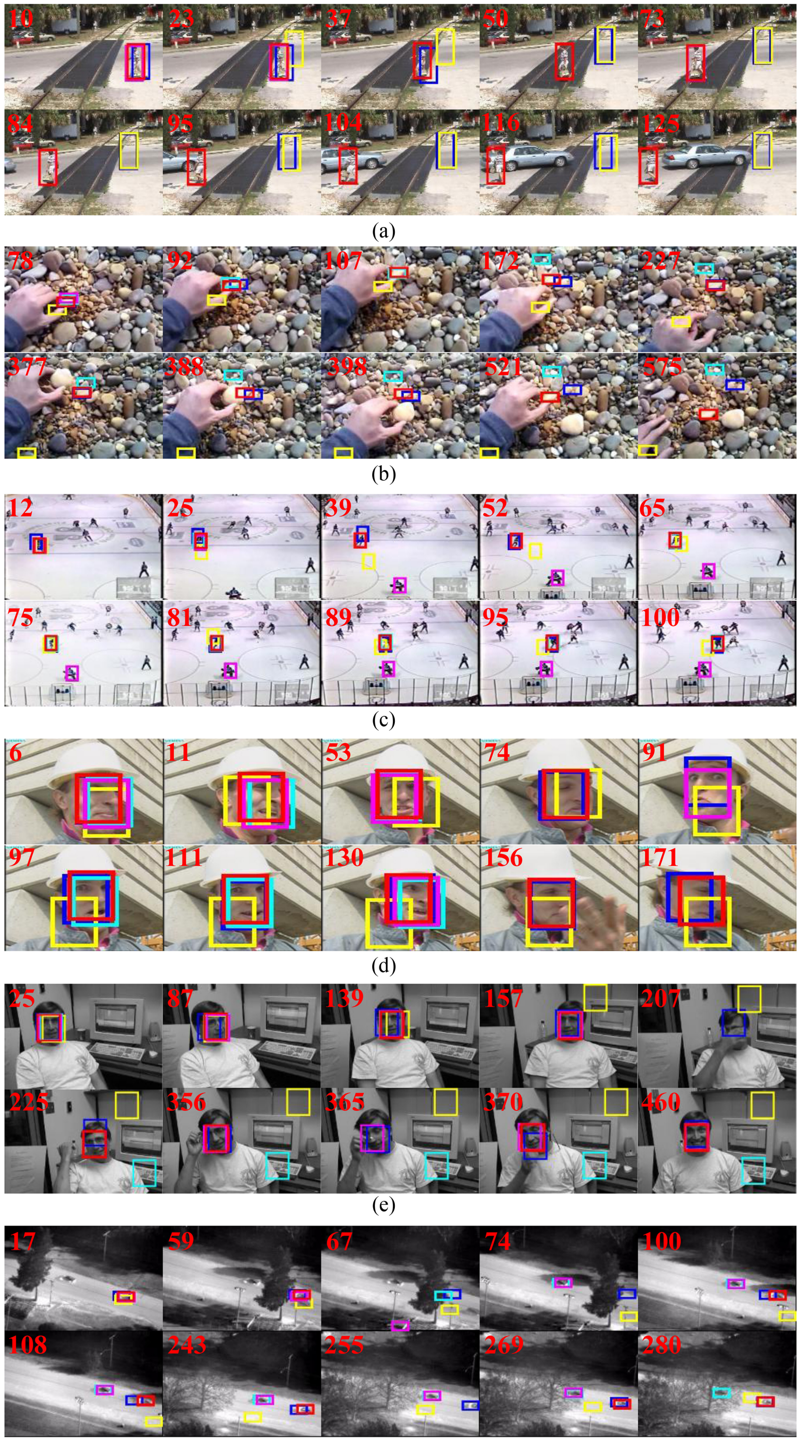


Fig. 6. Performance comparison of all the trackers (red: PRT; blue: MS; yellow: CT; cyan: SB; magenta: BSB).

cobblestone is moved, CT (yellow box) begins to locate the target inaccurately. Moreover, BSB (magenta box) and SB (cyan box) fail to locate the target from frames 92 and 172, respectively. In frames 388–398, the object is occluded by a bigger cobblestone moving over it, which leads to the tracking failure of MS (blue box). Comparatively, our PRT (red box) exhibits better discriminative ability and outperforms other baselines in the whole video. The plot of position error with respect to #frame is presented in Fig. 7(b), which demonstrates that the result of PRT is very close to the groundtruth.

D. Handling Large Appearance Change

The appearance change of a target is sometimes a noteworthy obstacle for achieving robust tracking, because under this situation the tracker will perhaps not identify the previous target any more. In the *Hockey*[5] sequence, a hockey player skates quickly to defend the attacker, and his body varies dramatically all the time. The performances of five trackers are compared in Fig. 6(c). When the player bends down to

[5]http://www.vision.ee.ethz.ch/ hegrabne/

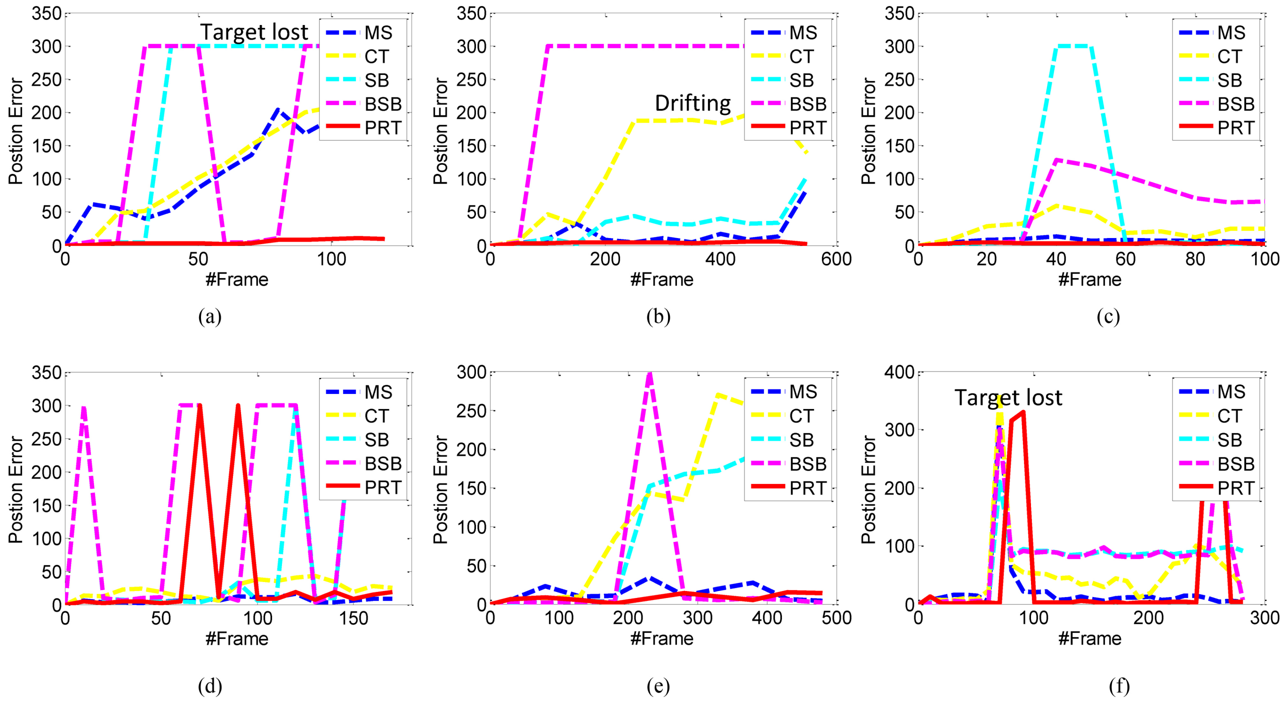


Fig. 7. Comparisons of position error. (a) *Pederstrain2* sequence. (b) *Stone* sequence. (c) *Hockey* sequence. (d) *Foreman* sequence. (e) *Dudek* sequence. (f) *Car* sequence.

accelerate (frames 25–100), the tracking box of CT (yellow box) drifts away form the target, SB (cyan box) cannot decide the position of the target, and BSB (magenta box) even locates the wrong target and switches to track the goalkeeper. Fig. 7(c) also illustrates the tracking failures of above methods. Comparatively, MS (blue box) and PRT (red box) perform better on this sequence.

E. Face Tracking Simulations

Face tracking is a popular research topic due to a wide variety of applications such as identity recognition, video conferencing, human-computer interaction, etc. Fig. 6(d) and (e) presents the performances of trackers on two public sequences *Foreman*[6] and *Dudek* respectively.

The sequence *Foreman* includes different head gestures, severe facial expression conversion and partial occlusion by hand. After a series of vigorous head movements, the yellow tracking box of CT drifts to somewhere else (frames 91–171); BSB (magenta box) cannot locate the target in many frames such as in 74, 111, 156 and 171. SB (cyan box) fails to decide the position of target in frames 74, 91, 156 and 171 because the appearance of the object changes drastically. Fig. 7(d) indicates that PRT also loses target around frame 90 due to the view change. Generally speaking, MS (blue box) and PRT (red box) can resist these disturbances and produce relatively better results than other trackers.

In *Dudek* sequence, during the frames 25–157 when the face turns from profile to front gradually, tracking box of CT drifts away from target. After that, the face is nearly completely occluded by hand, so MS can hardly implement tracking precisely. Note that our PRT is also unable to locate the target at this time because $score(\mathbf{x}_{top})$ is lower than the adaptive threshold as shown in (20), so no red tracking box appears in the frame 207. In the practical implementation of PRT, when the target is lost in current frame, the searching region in the next frame is enlarged in order to facilitate redetection. As a result, PRT successfully detects the target again after occlusion (frame 225), while both MS and SB fail to track the target precisely. During frames 365–370 when the object takes off the glasses, our algorithm can also detect the target vanishing correctly (frame 365). Fig. 7(e) plots the position error of all the trackers, which suggests that PRT achieves the best performance. For the visualization purpose, $score(\mathbf{x}_{top})$ in every frame is defined as a confidence level, and the variation of confidence level during the whole sequence is plotted in Fig. 8(a). The plot reveals that the confidence drops to relatively low level during frames 200–220 and 360–380. These two periods exactly correspond to the complete occlusion by hand and glasses removing [Fig. 6(e)]. It suggests that the score defined by (17) is reasonable, because it can accurately illustrate perturbations and effectively prevent tracking errors.

F. Dealing with Thermal IR Sequence

Thermal IR image is often used in surveillance applications at night, of which the image resolution is usually very low. In the sequence *Car*[7] two similar cars run toward the same direction, and the second one is the target as Fig. 6(f) shows. All trackers perform well before the target is occluded by a tall tree (frame 17). However, after occlusion MS (blue box) and CT (yellow box) begin to drift from target. SB (cyan box) and BSB (magenta box) locate the wrong target and begin to track the first car (frames 59–74). We think the reason is that the

[6]http://media.xiph.org/video/derf/

[7]http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html

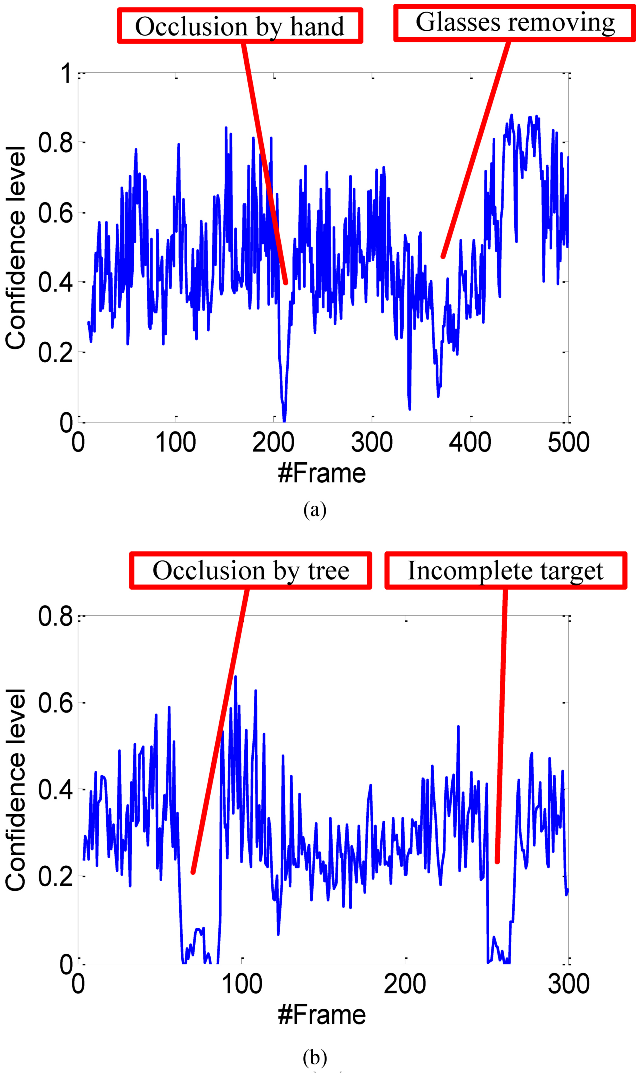


Fig. 8. Confidence level of PageRank tracker. (a) *Dudek* sequence. (b) *Car* sequence.

appearances of two cars are so similar that SB and BSB cannot make a clear distinction between the two. After the occlusion, only PRT (red box) finds the second car quickly and accurately (frames 108–243). Above is the first disturbance in this sequence. The second disturbance occurs around the frame 255 when part of the target is clipped by the edge of image. However, as soon as the entire car reappears, PRT locates it quickly and continues to track. This process is revealed by frames 269–280 in Fig. 6(f) and Fig. 7(f). The periods when confidence level drops dramatically in frames 50–90 and 260–280 [Fig. 8(b)] correspond to the two disturbances described above.

G. Discussion

The experiments presented in Section VII demonstrate that our PageRank tracker is more robust and adaptive than the remaining methods used for comparison. Our algorithm is able to handle complicated disturbances during tracking process, such as sudden background change, object appearance change, low image quality, partial or complete occlusion, etc. PageRank tracker also shows perfect recognition or redetection ability when target reappears after complete occlusion. These merits stem from the following reasons:

1) Ranking idea is adopted to choose the best region to represent the target.
2) Online learning process is incorporated into PageRank tracker, so that the tracker can know the latest appearance of target timely. Moreover, the graph is established by considering both local and pairwise information, so the relationship among samples can be expressed more precisely.
3) The methodology of filtering out the false positive samples ensures that each update of graph G is accurate, and this is very critical for avoiding drifting and achieving adaptive tracking.

VIII. Conclusion

This paper proposed a new graph-based tracking methodology by regarding visual tracking as a ranking problem. The PageRank algorithm, which is originally designed for ordering webpages, is adopted to retrieve the target region from massive unlabeled samples. The proposed PageRank graph not only incorporates the pairwise information among samples, but also exploits the local information of every node in the graph. Besides, PageRank vector is computed through the modified power method, which fuses the historical record of target into the original framework. Finally, we define a confidence score for the top-ranked samples, based on which a simple yet effective method for choosing precise target was developed. Our PageRank tracker was compared with some popular baselines and its performance was evaluated by different kinds of experiments from both qualitative and quantitative aspects. The proposed tracker performed encouraging results in tracking moving objects with sudden background or large appearance change, multiview face with severe expression change, and infrared target with full occlusion, etc. In this sense, PageRank tracker is very effective for robust and stable tracking.

A number of open issues remain. Practically, if more proper features are fused into our tracking framework, the performance is expected to be further improved; algorithmically, current conduction of PageRank tracker is computational expensive (3.05 fps for a 20×32 target), so efficient algorithms to implement PageRank tracker are in demand; and theoretically, whether other popular ranking methodologies can be applied to tracking is still to be investigated.

Appendix
Proof of Convergency

This section aims to prove that (11) converges to stationary PageRank vector $\mathbf{f}^{*T}$. According to (11)

$$\begin{aligned}
\mathbf{f}^{(t)T} &= \theta\mathbf{f}^{(t-1)T}\overline{\mathbf{P}} + (1-\theta)\,\mathbf{f}^{(0)T} \\
&= \theta\left[\theta\mathbf{f}^{(t-2)T}\overline{\mathbf{P}} + (1-\theta)\,\mathbf{f}^{(0)T}\right]\overline{\mathbf{P}} + (1-\theta)\,\mathbf{f}^{(0)T} \\
&= \ldots \\
&= \theta^t\mathbf{f}^{(0)T}\overline{\mathbf{P}}^t + (1-\theta)\,(\theta^{t-1}\mathbf{f}^{(0)T}\overline{\mathbf{P}}^{t-1} + \theta^{t-2}\mathbf{f}^{(0)T}\overline{\mathbf{P}}^{t-2} \\
&\quad + \cdots + \theta\mathbf{f}^{(0)T}\overline{\mathbf{P}} + \mathbf{f}^{(0)T}) \\
&= \underbrace{\mathbf{f}^{(0)T}\left(\theta\overline{\mathbf{P}}\right)^t}_{part\ 1} + \underbrace{(1-\theta)\sum_{i=0}^{t-1}\mathbf{f}^{(0)T}\left(\theta\overline{\mathbf{P}}\right)^i}_{part\ 2}.
\end{aligned} \tag{21}$$

Note that $\overline{\mathbf{P}}$ is a positive matrix, and the entries in every row of $\overline{\mathbf{P}}$ satisfy

$$\sum_{j=1}^{n} \overline{p}_{ij} = 1.$$

According to Perron-Frobenius theorem [38], the $\overline{\mathbf{P}}$'s Perron-Frobenius eigenvalue $r_{\overline{P}}$ equals to 1, hence the spectral radius of $\overline{\mathbf{P}}$, i.e. $\rho\left(\overline{\mathbf{P}}\right) = 1$. Besides, we have limited $\theta \in (0, 1)$, thus

$$\lim_{t\to\infty} \left(\theta\overline{\mathbf{P}}\right)^t = \mathbf{0}$$

$$\lim_{t\to\infty} \sum_{i=0}^{t} \mathbf{f}^{(0)T} \left(\theta\overline{\mathbf{P}}\right)^i = \mathbf{f}^{(0)T} \left(\mathbf{I} - \theta\overline{\mathbf{P}}\right)^{-1}$$

where $\mathbf{I}$ is the identity matrix of the same size as $\overline{\mathbf{P}}$, so both part 1 and part 2 in (21) are convergent. Therefore, the sequence $\{\mathbf{f}^{(t)T}\}$ produced by (11) finally converges to

$$\mathbf{f}^{*T} = \lim_{t\to\infty} \mathbf{f}^{(t)T} = (1-\theta)\,\mathbf{f}^{(0)T}\left(\mathbf{I} - \theta\overline{\mathbf{P}}\right)^{-1}. \tag{22}$$

REFERENCES

[1] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[2] J. Schmudderich, V. Willert, J. Eggert, S. Rebhan, C. Goerick, G. Sagerer, and E. Korner, "Estimating object proper motion using optical flow, kinematics, and depth information," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1139–1151, Aug. 2008.

[3] 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.

[4] J. Wang and Y. Yagi, "Adaptive mean-shift tracking with auxiliary particles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1578–1589, Dec. 2009.

[5] M. Isard and A. Blake, "Condensation—Conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[6] A. Kiruluta, M. Eizenman, and S. Pasupathy, "Predictive head movement tracking using a Kalman filter," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 2, pp. 326–331, Apr. 1997.

[7] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers, "A two-stage dynamic model for visual tracking," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1505–1520, Dec. 2010.

[8] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. CVPR*, 2009, pp. 983–990.

[9] B. Babenko and M. Y. S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2010.

[10] H. Lu, Q. Zhou, D. Wang, and R. Xiang, "A co-training framework for visual tracking with multiple instance learning," in *Proc. IEEE Int. Conf. Automat. Face Gesture Recognit. Workshops*, 2011, pp. 539–544.

[11] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognit.*, vol. 46, no. 1, pp. 397–411, 2013.

[12] Z. Ni, S. Sunderrajan, A. Rahimi, and B. Manjunath, "Particle filter tracking with online multiple instance learning," in *Proc. IEEE ICPR*, 2010, pp. 2616–2619.

[13] X. Zhu and B. Goldberg, "Introduction to semi-supervised learning," in *Synthesis Lectures On Artificial Intelligence And Machine Learning*. San Rafael, CA, USA: Morgan & Claypool, 2009.

[14] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. 11th IEEE ICCV*, 2007, pp. 1–8.

[15] C. Gong, Y. Liu, T. Li, J. Yang, and X. He, "The extended co-learning framework for robust object tracking," in *Proc. IEEE Int. Conf. ICME*, 2012, pp. 398–403.

[16] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th ICML*, 2003, pp. 912–919.

[17] Y. Zha, Y. Yang, and D. Bi, "Graph-based transductive learning for robust visual tracking," *Pattern Recognit.*, vol. 43, no. 1, pp. 187–196, 2010.

[18] C. Gong, K. Fu, E. Tu, J. Yang, and X. He, "Robust object tracking using linear neighborhood propagation," *J. Electron. Imag.*, vol. 22, no. 1, 2013.

[19] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *Proc. ACM 23rd ICML*, 2006, pp. 985–992.

[20] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. CVPR*, vol. 1, 2006, pp. 260–267.

[21] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. ECCV*, pp. 234–247, 2008.

[22] S. Stalder, H. Grabner, and L. Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Proc. IEEE ICCV Workshops*, 2009, pp. 1409–1416.

[23] Z. Kalal, J. Matas, and K. Mikolajczyk, "PN learning: Bootstrapping binary classifiers by structural constraints," in *Proc. Conf. on IEEE CVPR*, 2010, pp. 49–56.

[24] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: Tracking-learning-detection applied to faces," in *Proc. 17th IEEE ICIP*, 2010, pp. 3789–3792.

[25] H. Liu and F. Sun, "Semi-supervised particle filter for visual tracking," in *Proc. IEEE ICRA*, 2009, pp. 3928–3933.

[26] H. Liu and F. Sun, "Semi-supervised ensemble tracking," in *Proc. IEEE ICASSP* , 2009, pp. 1645–1648.

[27] H. Wang, X. Hou, and C. Liu, "Boosting incremental semi-supervised discriminant analysis for tracking," in *Proc. 20th IEEE ICPR*, 2010, pp. 2748–2751.

[28] B. Zhong, H. Yao, S. Chen, R. Ji, X. Yuan, S. Liu, and W. Gao, "Visual tracking via weakly supervised learning from multiple imperfect oracles," in *Proc. IEEE Conf. CVPR*, 2010, pp. 1323–1330.

[29] J. Kwon and K. Lee, "Tracking by sampling trackers," in *Proc. IEEE ICCV*, 2011, pp. 1195–1202.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," 1999.

[31] C. Yang and K. Chan, "Retrieving multimedia web objects based on PageRank algorithm," in *Proc. Spl. Interest Tracks Posters 14th Int. Conf. World Wide Web*, 2005, pp. 906–907.

[32] K. Avrachenkov, V. Dobrynin, D. Nemirovsky, S. Pham, and E. Smirnova, "Pagerank based clustering of hypertext document collections," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2008, pp. 873–874.

[33] A. Langville and C. Meyer. (2003). "Fiddling with PageRank," *Center for Research in Scientific Computation, North Carolina State Univ., Rep. CRSC-TR03-34*. [Online] Available: http://meyer. math. ncsu.edu/Meyer/PS_Files/FiddlingPageRank.pdf

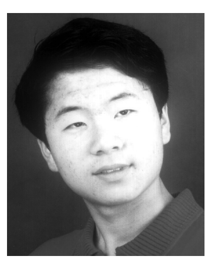
[34] B. Walsh. (2004). "Markov chain Monte Carlo and Gibbs sampling," *Lecture Notes for EEB 581*. [Online] Available: http://web.mit.edu/ wingated/www/introductions/mcmc-gibbs-intro.pdf

[35] T. Haveliwala, "Efficient computation of PageRank," Stanford Univ., CA, USA, Tech. Rep 1999-31, 1999.

[36] Z. Yin and R. Collins, "Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking," in *Proc. IEEE Conf. CVPR*, 2008.

[37] T. Lee, "Image representation using 2d gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, Oct. 1996.

[38] G. Golub and C. Loan, *Matrix Computations*. California, MD, USA: Johns Hopkins Univ. Press, 1996.



Chen Gong received the B.S. degree in control science and engineering from East China University of Science and Technology, Shanghai, China, in 2010. He is currently pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China.

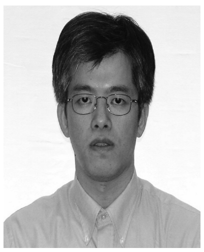
His current research interests include machine learning, object detection, and tracking.

Keren Fu received the B.Sc. degree in automation from Huazhong University of Science and Technology, Hubei, China, in 2011. Currently, he is pursuing the Ph.D. degree at the Image Processing and Pattern Recognition Laboratory, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include object detection, saliency detection, visual tracking, and machine learning.

Artur Loza received the M.Sc. degree in electrical engineering from Wroclaw University of Technology, Wroclaw, Poland, in 2000, and the Ph.D. degree from the University of Bristol, Bristol, U.K., in 2006.

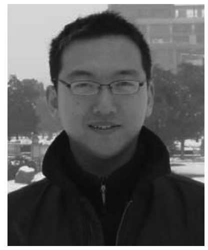
He has also been a Research Associate at the University of Bristol. In 2010, he was awarded an EU-funded two-year scholarship to conduct his research at Shanghai Jiao Tong University, Shanghai, China. His current research interests include statistical and multiresolution signal and image processing, multimodal image fusion, and Bayesian-based video object detection and tracking.



Qiang Wu (M'03) received the B.Eng. and M.Eng. degrees in electronic engineering from the Harbin Institute of Technology, Harbin, China, in 1996 and 1998, respectively, and the Ph.D. degree in computing science from the University of Technology Sydney, Sydney, Australia, in 2004.

He is currently a Senior Lecturer with the School of Computing and Communications, University of Technology, Sydney. He has published more than 70 refereed papers, including those published in prestigious journals and top international conferences. His current research interests include computer vision, image processing, pattern recognition, machine learning, and multimedia processing.

Dr. Wu has been a Guest Editor of several international journals, such as *Pattern Recognition Letters* and the *International Journal of Pattern Recognition and Artificial Intelligence*. He has served as a Chair and/or a Program Committee member for a number of international conferences. He has also served as a Reviewer for several international journals, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B: CYBERNETICS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Pattern Recognition*, *Pattern Recognition Letters*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *EURASIP Journal on Image and Video Processing*.



Jia Liu received the B.E. and M.S. degrees from the Engineering College of Armed Police Force, Xi'an, China.

He is currently pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include computer vision, machine learning, and visual surveillance.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University, Hamburg, Germany, in 1994.

Currently, he is a professor at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 200 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.