Pattern Recognition 44 (2011) 2170–2183

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Visual object tracking via sample-based Adaptive Sparse Representation (AdaSR)

Zhenjun Han[a], Jianbin Jiao[a,*], Baochang Zhang[b], Qixiang Ye[a], Jianzhuang Liu[c]

[a] *Graduate University of Chinese Academy of Sciences, No. 19A, Yu Quan Road, Shi Jing Shan District, 100049 Beijing, PR China*
[b] *Beijing University of Aeronautics and Astronautics, China*
[c] *Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong*

ARTICLE INFO

Article history:
Received 2 August 2010
Received in revised form
8 January 2011
Accepted 3 March 2011
Available online 11 March 2011

Keywords:
Object tracking
Sample-Based Representation
Adaptive sparse representation

ABSTRACT

When appearance variation of object and its background, partial occlusion or deterioration in object images occurs, most existing visual tracking methods tend to fail in tracking the target. To address this problem, this paper proposes a new approach for visual object tracking based on Sample-Based Adaptive Sparse Representation (AdaSR), which ensures that the tracked object is adaptively and compactly expressed with predefined samples. First, the Sample-Based Sparse Representation, which selects a subset of samples as a basis for object representation by exploiting L1-norm minimization, improves the representation adaptation to partial occlusion for tracking. Second, to keep the temporal consistency and adaptation to appearance variation and deterioration in object images during the tracking process, the object's Sample-Based Sparse Representation is adaptively evaluated based on a Kalman filter, obtaining the AdaSR. Finally, the candidate holding the most similar Sample-Based Sparse Representation to the AdaSR of the tracked object will be regarded as the instantaneous tracking result. In addition, we can easily extend the AdaSR for multi-object tracking by integrating the sample set of each tracked object (named Common Sample-Based Adaptive Sparse Representation Analysis (AdaSRA)). AdaSRA fully analyses Adaptive Sparse Representation similarity for object classification. Our experiments on public datasets show state-of-the-art results, which are better than those of several representative tracking methods.

1. Introduction

Research on object tracking in the past decades has yielded an arsenal of powerful algorithms, which play important roles in many applications, such as automatic visual surveillance [1], human computer interaction systems [2] and robotics [3], where moving objects in videos with stationary or even dynamic background can be effectively tracked, given simple data association techniques.

Although researchers have made great progresses, there still exist many open problems facing object motion state variation, the appearance variation of either object or background and the occlusions, etc. The previous research on object tracking focused on template feature extraction and matching, which includes three different categories: motion models [4,5], searching methods [6,7] and object representation [8–18].

Motion models are employed to predict the object's location in a new frame within a video sequence based on its history motion characteristics. This can improve the tracking stabilization and make the tracking survive some occlusions if the trajectories of objects are correctly predicted. Early works used a Kalman filter [4] to provide solutions that are optimal for a linear Gaussian model. The particle filter, also known as the sequential Monte Carlo method, is one of the most popular approaches. It recursively constructs the posterior probability density function of the state space using Monte Carlo integration. It has been developed in the computer vision community and applied to tracking problems under the name of Condensation [5].

Searching methods are also indispensable to a successful tracking. Given a tracked object, searching methods use various matching strategies to find its location in a new video frame. In addition, when the object varies in size, it is needed to calculate the scale parameter. Early work [6] used the sum of squared difference (SSD) as a cost function in the tracking problem. Subsequently, more robust similarity measures have been applied such as the mean-shift algorithm [7] was utilized to find the optimal solution.

Although motion models and searching algorithms are crucial to object tracking, it is not true that a proper motion model together with a good searching algorithm will always lead to good

[*] Corresponding author. Tel.: +86 10 88256968.
E-mail address: jiaojb@gucas.ac.cn (J. Jiao).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.03.002

tracking results. Therefore, in this paper, we mainly investigate the related problem: ‘‘What is an effective object representation for tracking when there will be unpredictable appearance variation of the object and its background, partial occlusion and deterioration in object images?’’

2. Related works

2.1. ‘‘Many could be better than one’’

Object representation is a key part of object tracking. In the existing approaches, most algorithms model the object appearance by extracting features from the global object region [9–13]. Color histogram is one of the most widely used feature [13,14] for its effectiveness and efficiency. Some other characteristics, such as texture, contour and feature point features, are employed to represent the object [15,16]. Furthermore, combinations of these features are proposed for representation [10–12]. In addition, due to the appearance variations of tracked object and its background during the tracking process, there have been enormous efforts on finding the ‘‘optimal’’ features for discriminating the object with its background. Collins et al. [13] online selected the top M most discriminative features for tracking. In [11], Han et al. proposed combined feature evaluation in Kalman filter frameworks for adaptive object tracking. In [12], an appearance-adaptive model is incorporated in a particle filter to realize robust visual tracking and classification algorithms.

In [17–20], researchers proposed to segment a tracked object into local regions and track the local regions individually to improve tracking performance. Actually, they model the object appearance as a bag of local regions, and treats object tracking as individually tracking each part of the object. In [20], they proposed a dynamic spatial bias appearance model based on online learning the spatial bias of the object appearance dynamically using local region confidences to guide object tracking in cases of appearances variations of object and background.

The above tracking methods, which depend on global or local template feature extraction and matching as shown in Fig. 1 (the top row), achieve some success, however, they always suffer from tracking failures caused by template drift, which often occurs in a long duration tracking.

2.2. ‘‘Many could be better than all’’

Considering the basic issue of a tracking problem is to locate a specific object in a searching area in a new frame, it is reasonable to make a hypothesis that the tracked object can be represented as a linear superposition of the samples just inside the searching area. The searching area with careful definition is always much larger than the tracked object region in tracking process, and subsequently leads to some informative background samples for superposition. Sometimes this could be necessary and efficient for modeling the appearance variations of the object caused by partial occlusion or deterioration in object images, since it is nearly impossible to predict or model the appearance evolution process only based on the object itself.

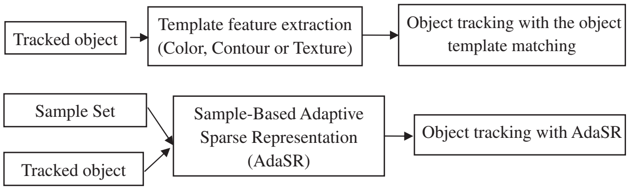

Tracked object
Template feature extraction (Color, Contour or Texture)
Object tracking with the object template matching
Sample Set
Tracked object
Sample-Based Adaptive Sparse Representation (AdaSR)
Object tracking with AdaSR

Fig. 1. The top row is the flowchart of the template feature extraction and matching based tracking method; the bottom row is the flowchart of the proposed object tracking method.

However, because of not enough prior knowledge, a long-term suitable searching area for object tracking during the tracking process is nearly impossible. Therefore, it is inevitable that the samples from background are not all informative and useful for representing the object, that is to say, there are many redundant samples for linear superposition. It is necessary for us to make some sample selection.

What’s more, modern investigation in the human vision system (HVS) has shown that a selective small subset of neurons is active for a variety of specific stimuli [21], such as color, texture, shape, and scale. Among the large amount of neurons in the human vision system, the firing of neurons to a specific object is typically highly sparse. Based on the study of HVS, sparse representation of an object has been brought out [22]. In [22] and some other existing researches [23–25], all of them showed a common sense: using parsimony as the principle for choosing a limited subset of samples from a set, rather than directly using all the samples, for representing an object is more effective.

2.3. Motivation

Inspired by the basic issue of object tracking and the development of sparse representation for object representation, in this paper, we cast the tracking as finding a sparse representation of the tracked object based on a dynamically constructed and updated sample set during the tracking process as shown in Fig. 1 (the bottom row). This provides a new way to solve object deterioration and the partial occlusion problems compared with the traditional tracking approaches. When combining with L1-norm minimization for Sample-Based Sparse Representation, intuition behind which lies in the fact that some coefficients of the Sparse Representation compactly expressing an object are nonzero and other coefficients are almost zero. And considering that the procedure of tracking is always temporal consistency, we online evaluate Sample-Based Sparse Representation of the tracked object in a Kalman filter by exploiting the Sparse Representation in the current frame and those in the previous frames. Finally, we investigate the property of Sample-Based Adaptive Sparse Representation, which is effective as its coefficients are the discriminative information between objects, therefore, the candidate holding the most similar coefficient vector with the tracked object will be seen as the instantaneous tracking result. Different from [5], in which the sample set is used to model the motion state of the tracked object, our sample set is used to extract sparse representation of the object. In this paper, our method is different from [26] mainly in the evolution of sparse representation based on the sample set. We dynamically and adaptively evaluate the sparse representation in filter framework, which can ensure the temporal adaptation and consistency of the appearance variations and lead to more robust tracking results. Comparisons in details will be given in Section 5.

In addition, the Sparse Representation Classifier (SRC) [22] based on residuals is very effective on face recognition, when there are enough training samples. As a robust multi-class classifier, SRC can also be used to solve the data association problem in multi-object tracking by classifying each tracked object. However, the effectiveness of SRC severely drops when there are not enough samples for tracked objects. To solve this problem, we fully utilize the property of AdaSR, and then propose an extended method named the Sample-Based Adaptive Sparse Representation Analysis (AdaSRA), which enriches its discriminative ability by reserving all coefficients for measuring the relationship between an object and each class. The extensive experiments in Section 5.3 also show the superiority of AdaSRA over SRC. Compared with classical methods, like joint probabilities data association filter (JPDAF) [27], the intuition behind AdaSRA still lies in the fact that the coefficients can be used to discriminate between various classes based on a common basis, which is the core of the data association between multiple objects in the tracking process.

The rest of the paper is organized as follows. Single Object Tracking via Sample-Based Adaptive Sparse Representation is described in Section 3. Multi-object Tracking via Sample-Based Adaptive Common Sparse Representation Analysis is discussed in Section 4. Experiments with comparisons are given in Section 5. Section 6 concludes this paper.

3. Object tracking via Sample-Based Adaptive Sparse Representation

Firstly, a sample set for the tracked object is constructed at the beginning of a tracking process. Then, we represent the object in a sparse and adaptive way during the tracking process based on the sample set, where the Sample-Based Adaptive Sparse Representation of the object is first extracted by calculating the L1-norm minimization and then evaluated in Kalman filter framework. Finally, we track the object in a new video frame based on its AdaSR. In addition, after some tracking frames, we update the sample set and re-calculate the sparse representation (re-initiate the tracking process). The flowchart is shown in Fig. 2.

3.1. Sample set construction and updating

A sample set is constructed for the object based on a window, named sample window (the black square region in Fig. 3a), which is a sub-image centered around the object when the tracking is initialized. Each sample in the set is defined as a sub-window of the sample window (the red rectangle region in Fig. 3b). A sample rectangle in the window is specified by $r=(x,y,s,\alpha)$ with $0<x<W$, $0<y<H$, $s>0$, $0^0 \le \alpha \le 360^0$. This sample set is almost infinitely large. For practical reasons, it is reduced as follows:

1. The (x,y) varies with the step of d pixels in horizontal and vertical orientations.
2. The s is uniformly sampled from {0.8, 0.9, 1.0,1.1, 1.2} times of the tracked object's size, when (x,y) is fixed.
3. The α is set as 0° in our approach, for rotation of the object is not considered in current research.

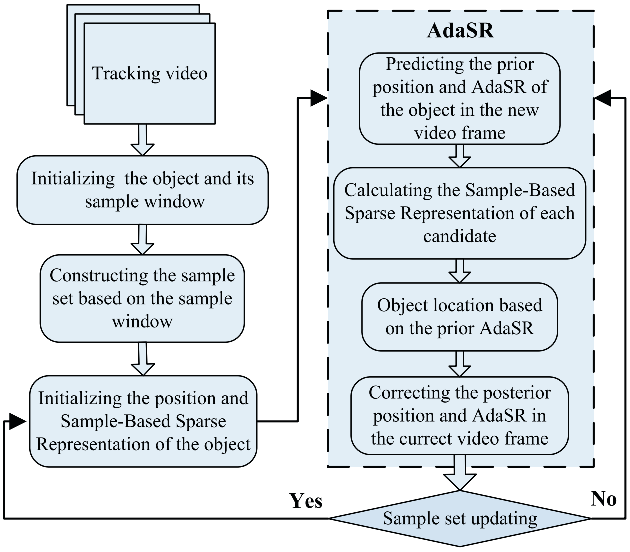

Tracking video
Initializing the object and its sample window
Constructing the sample set based on the sample window
Initializing the position and Sample-Based Sparse Representation of the object
AdaSR
Predicting the prior position and AdaSR of the object in the new video frame
Calculating the Sample-Based Sparse Representation of each candidate
Object location based on the prior AdaSR
Correcting the posterior position and AdaSR in the currect video frame
Yes
No
Sample set updating

Fig. 2. Flowchart of the proposed tracking method.

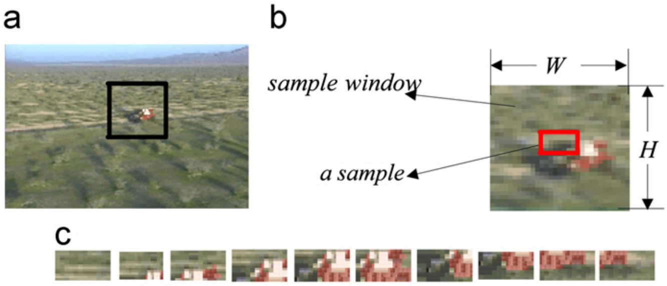


Fig. 3. (a) The sample window of the object, (b) a sample, and (c) examples in the sample set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In our experiments, the sample window is empirically set as the searching area (9 or 16 times of the size of the object region), and $d=4$ These restrictions lead to reasonable number of samples in the set. Supposing we totally obtain K samples for constructing the set $\{s^k, k=1,\ldots,K\}$ for an object, where most of the samples are associated with the background, and a few of them are parts of or the whole object (shown in Fig. 3c).

The sample set should be updated in tracking instantaneously. In the tracking process, for each m tracking frames (m is set to 20 in our experiments), we randomly choose a sample from the set and replace it with the latest tracking result of the object. The updating of the sample set ensures that most recent object appearances are reflected in the sample set, which is another reason why our tracking is adaptive. It should be noted that only one sample in the set is replaced with the latest tracking result each time, and so even a bad sample replacement during the tracking process of the proposed updating strategy affects little on the whole set used for calculating the adaptive sample-based sparse representation, which avoids the drift problem and ensures tracking stability. While in most of existing approaches like [28], a bad template updating method tends to cause the template drift problem and then leads to tracking failures, especially when there are object appearance variations or occlusions.

We use the Histogram of Colors (HC) in the RGB color space and the Histogram of Oriented Gradients (HOG) on the gray-level image to calculate the combined feature set (named the Histogram of Oriented Gradients and Colors (HOGC)) to represent the tracked object and the samples in the set. Then we can obtain a feature set $A=\{a^k,\ k=1,\ldots,K\}$ and F for the sample set and the object, respectively. The extraction of the combined feature is described as follows.

3.1.1. Histogram of color

To calculate the color histogram, which is generally robust to rotation and deformation [29], the RGB color space is chosen for its simplicity. We first convert the color information of each pixel into a quantized value, and then the quantized value is mapped to an index of a corresponding histogram bin. The number of pixels assigned to each bin is accumulated over the whole image patch. In this paper, each color component (R, G and B) is linearly quantized into 16 levels and then a histogram of 16 dimensions is extracted on each component. We obtain a HC of 48 dimensions totally.

3.1.2. Histogram of oriented gradient

Motivated by the work in [30], a histogram of 72 dimensions is extracted to describe the gradient orientation of a rectangle region, called HOG. Details of HOG feature extraction are described as follows.

HOG is calculated in grayscale space. We first resize the rectangle region of object or each sample into a normalized window of fixed size, say 32 × 32 pixels. Then, we divide the window into small spatial cells with the size of 8 × 8 and 4 (2 × 2) such adjacent cells are then integrated into a block, therefore we can obtain 9 blocks numbered from ① to ⑨, which overlap each other (shown in Fig. 4a). Each pixel in a block calculates its gradient orientation $ori(h,w)$ based on Eq. (1). The mask for the calculation of $ori(h,w)$ is shown in Fig. 4b. Different from the method in [30], each block in this approach constructs an 8-bin HOG without local normalization (shown in Fig. 4c). Then we combine the HOG of each block to obtain a 72-dimension feature for the rectangle region of object or each sample

$$dy = I(h+1,w) - I(h-1,w), dx = I(h,w+1) - I(h,w-1), ori(h,w) = a\tan 2(dy,dx) \quad ori \in [-\pi,\pi] \tag{1}$$

3.2. Object sparse representation based on the sample set

When we obtain the sample set, it is reasonable to make hypothesis that the tracked object can be approximately represented as a linear superposition of the samples just inside the sample window as follows:

$$A\psi \approx F \tag{2}$$

where F is the feature vector of the object, $\psi = \{\psi^k, k = 1,\ldots,K\}$ is a coefficient vector for superposition associated with A and F, and ψ^k is the coefficient of the kth sample in the set. Although the above model can also be more complicated, we assume a linear system in our paper from both efficiency requirement of a practical application and simplicity of representation.

In a real condition, the sample window is always much larger than the tracked object region, leading to a sparse coefficient vector of the linear superposition, since there may be many redundant samples in the set. In the case of partial occlusion, a limited number of negative samples and some positive samples (samples obtained from the tracked object region, such as parts of or the whole object) will be activated, but the whole coefficient vector remains sparse, supposing there are r nonzero coefficients in $\psi = \{\psi^k, k = 1,\ldots,K\}$, we can reasonably infer that $r \ll K$. In this case, we say that the object has an *r*-sparse representation based on the sample set. The number of the nonzero coefficients is denoted by $\|\psi\|_0$. Minimizing $\|\psi\|_0$ is the principle to obtain a sparse representation, which is, however, an NP-hard problem. Recent development in the theory of compact sensing [31] shows that the solution of L1-norm minimization subject to a linear system of the samples can be used to find sparse enough representation of the object. The resulting optimization problem, similar to the LASSO in statistics [32], penalizes the L1-norm of the coefficients in the linear combination, rather than directly penalizing the number of nonzero coefficients ($\|\psi\|_0$). In terms of the set *A* and the object *F*, a sparse representation is computed as follows:

$$\arg\min \|\psi\|_1, \text{ subject to } A\psi = F, \tag{3}$$

where $\|\cdot\|_1$ represents the L1-norm.

Since real images are noisy, it may not be possible to precisely express the object directly with a sparse representation of the samples. To model the noise in the video frame, we empirically consider a noise term as $\varepsilon = 0.1$ for the tracked object, and Eq. (3) is then modified as

$$\arg\min \|\psi\|_1, \text{ subject to } \quad \|A\psi - F\|_2 \le \varepsilon \tag{4}$$

where $\|\cdot\|_2$ represents the L2-norm. This model can be solved in polynomial time by a linear programming or quadratic programming method [33]. Even more efficient methods are available when the solution is known to be very sparse. For example, Homotopy algorithm [34] runs much more rapidly than general-purpose LP solvers when sufficient sparsity is present. Indeed, the method often has the following k-step solution property: if the underlying solution has only k nonzero coefficients, the Homotopy method reaches that solution in only k iterative steps. When this property holds and k is small compared to the problem size, L1-norm minimization problems with k-sparse solutions can be solved in a fraction of the cost of solving one full-sized linear system.

By solving Eq. (4), the vector of r ($r \ll K$) sparse coefficients can be obtained. An example of the coefficient vector is given in Fig. 5, where we use 100 samples in the set for calculating its sparse coefficients; it is found that about 10 samples are selected to

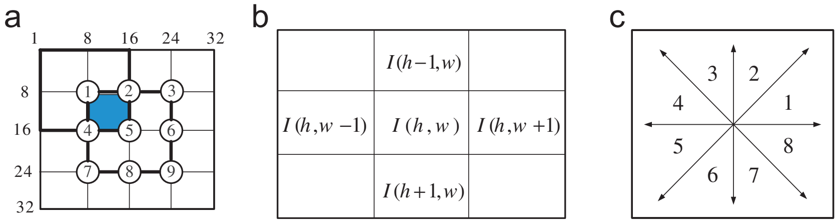


Fig. 4. HOG feature extraction: (a) 9 blocks for HOG feature extraction, (b) mask for pixel gradient calculation, and (c) orientation bins.

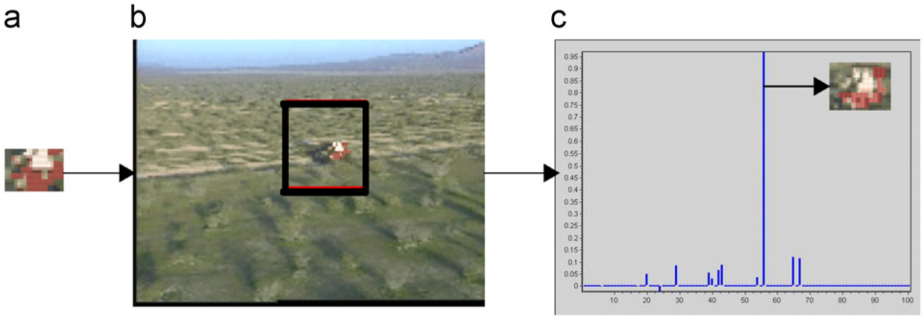


Fig. 5. (a) A tracked object, (b) the sample window, and (c) the sparse coefficient vector.

express the object with a very small reconstruction error. It can be seen from Fig. 5c that there are about 10 coefficients of the vector are nonzero, samples corresponding to which are of the most representative ability for the object in the tracking procedure, showing the high sparsity.

In terms of the sample set A and its corresponding coefficient vector ψ, the object is represented as follows when the tracking procedure is initialed:

$$\left\{(a^1,\psi^1),(a^2,\psi^2)\ldots(a^k,\psi^k)\ldots(a^K,\psi^K)\right\} \tag{5}$$

The property of the sparse representation can guarantee that the object is represented in the most compact way based on the sample set. That is to say, Sample-Based Sparse Representation selects a small subset, in which the samples are the most representative ones, such as parts of or the whole object.

3.3. *Object tracking based on Adaptive Sparse Representation (AdaSR)*

During the tracking process, since the appearances of the tracked object and its corresponding background are always variational, therefore, on one side we update the sample set and on another side we online adaptively model the evolution of the object's sparse representation in Kalman filter between updatings.

3.3.1. *Definition of the Kalman filter*

In this paper, we put forward a lumped model for the Kalman filter, which provides a recursive solution to the linear optimal filtering problem and applies to stationary as well as non-stationary environment [35], including the evolution of the object sparse representation and its motion model. The state of the Kalman filter is the sparse representation and the position of the object, while the measurements include the sparse representation and the location of the instantaneous tracking result. This can induce a Kalman filter as

$$\begin{cases} \begin{pmatrix} \widetilde{\psi_{t+1}} \\ \Delta\widetilde{\psi_{t+1}} \\ \widetilde{Pos_{t+1}} \\ \Delta\widetilde{Pos_{t+1}} \end{pmatrix} = \begin{pmatrix} I_{K\times K} & I_{K\times K} & 0 & 0 \\ 0 & I_{K\times K} & 0 & 0 \\ 0 & 0 & I_{M\times M} & I_{M\times M} \\ 0 & 0 & 0 & I_{M\times M} \end{pmatrix} \begin{pmatrix} \psi_t \\ \Delta\psi_t \\ Pos_t \\ \Delta Pos_t \end{pmatrix} + u_t \\ \begin{pmatrix} m\psi_t \\ mPos_t \end{pmatrix} = \begin{pmatrix} I_{K\times K} & 0 & 0 & 0 \\ 0 & 0 & I_{M\times M} & 0 \end{pmatrix} \begin{pmatrix} \psi_t \\ \Delta\psi_t \\ Pos_t \\ \Delta Pos_t \end{pmatrix} + v_t \end{cases} \tag{6}$$

where $\psi_t=\{\psi_t^k, k=1\ldots K\}$ is the posterior sparse coefficient vector at frame t, $\Delta\psi_t=\psi_t-\psi_{t-1}$, $m\psi_t$ is the instantaneous sparse coefficient vector of the tracking result at frame t, Pos_t is the posterior location of the tracked object at frame t, $\Delta Pos_t = Pos_t - Pos_{t-1}$, $mPos_t$ is the location obtained after the matching procedure, M is set to 2, u_t and v_t are both Gaussian white noises empirically, and $I_{K\times K}$ is an identity matrix in our experiment.

3.3.2. *Object searching with adaptive prior Sparse Representation*

When a new video frame t is coming, the tracking procedure is performed as an exhaustive search algorithm in the searching area Ω_t in the tth video frame. Our goal is to find the object location $(x,y)^*$ in Ω_t by minimizing the difference between the adaptive prior sparse representation of the object with the sparse representation of each candidate in Ω_t.

$$\operatorname*{Min}_{(x,y)\in\Omega_t}\left(\left\|\widetilde{\psi_t}-m\psi(C_t(x,y))\right\|_1\right) = \operatorname*{Min}_{(x,y)\in\Omega_t}\left(\left\|\sum_{k=1}^{K}\widetilde{\psi_t^k}-m\psi^k(C_t(x,y))\right\|_1\right) \tag{7}$$

where $C_t(x,y)$ is the candidate at location (x,y) in Ω_t, $m\psi(C_t(x,y))$ represents the sparse representation of candidate $C_t(x,y)$.

After we obtain the best match of the object in the searching area in frame t by Eq. (7), we carry out the state correction procedure based on the Kalman filter to obtain the posterior sparse representation and position of the tracked object. Algorithm 1 below summarizes the object tracking approach.

Algorithm 1: Object tracking via Adaptive Sparse Representation

```
1. Initialization ($t=0$).
   1.1 Initializing the tracked object $F$ manually;
   1.2 Constructing the sample set $A$ of the object;
   1.3 Initializing the sparse coefficient vector $\psi$ of the tracked
       object in terms of $A$and $F$ using the L1-norm minimization.
2. Object tracking ($t>0$). In a new video frame:
   2.1 Predicting the prior sparse representation and position of
       the tracked object in the new frame;
   2.2 Searching the object's location in the searching area $\Omega_t$ by
       minimizing the residual between prior sparse representation
       with the instantaneous sparse representation of each
       candidate in $\Omega_t$;
   2.3 Correcting the posterior sparse representation and the
       position.
3. $t=t+1$. If no updating, go to step 2 or end the tracking loop;
   otherwise go to step 4.
4. Sample set updating ($t>0$).
   4.1 Selecting a sample from the set randomly, and replace it
       with the latest tracking result;
   4.2 Go to step 1.2.
```

4. Multi-object tracking via Sample-Based Adaptive Sparse Representation analysis

In a multi-object tracking system, precisely tracking each single object is necessary but not enough, since given an object in a multi-object tracking system, we should still know "which object we are tracking?"

We propose a novel scheme to assign an instantaneous tracked result to a class based on adaptive sparse representation analysis (AdaSRA). For multi-object tracking, we construct a common sample set at the beginning of the tracking process, by combining the sample set for each object in the following Eq. (8). We update the common sample set by updating the sample set of each object as described in Section 3.1

$$B=\cup\{A_j\}=\{a_1^1,a_1^2,\ldots,a_1^K,\ a_2^1,a_2^2\ldots a_2^K,\ldots,a_J^K\},\quad j=1,\ldots,J \tag{8}$$

where J is the number of all tracked objects, A_j is the feature set of all the samples of the jth object. If there are K samples in each object's sample set, there are total $K\times J$ samples in the common set. We can calculate the common sparse representation for each object based on B by exploiting the L1-norm minimization with Eq. (4).

At the beginning of the tracking process, for each initialized object, say, the jth object, we calculate its common AdaSR $\psi_{0,j}$, which is named the reference AdaSR of the jth object based on the common sample set B. In addition, considering the appearance variations of each object, we online adaptively evaluate their reference AdaSRs $(\psi_{0,j})$ based on Eq. (6), which ensures that the evolution of each reference AdaSR is temporally consistent. During the tracking process, when we obtain an instantaneous tracked object F_{unknow} with its corresponding common sparse representation $\psi_{t,unknow}$ at frame t, we assign the tracked object to a class by calculating the similarity between the instantaneous common sparse representation with each prior reference AdaSR as follows:

$$\mathrm{Classify}(F_{unknow}) = \underset{j}{\mathrm{argmin}}\, r_j(F_{unknow}) \quad \text{where} \quad r_j(F_{unknow}) = \|\psi_{t,unknow} - \widetilde{\psi_{0,j}}\|_2 \tag{9}$$

Our scheme is different from SRC [22], in which, for each class j, they define its characteristic function δ_j, which only selects the coefficients associated with the jth class. For $\psi_{t,unknow}$, $\delta_j(\psi_{t,unknow})$ is a new vector, whose nonzero entries are the ones in $\psi_{t,unknow}$ that are associated with class j, and whose entries associated with other classes are zero as

$$\mathrm{Classify}\ (F_{unknow}) = \underset{j}{\mathrm{argmin}}\, r_j(\mathrm{F}_{unknow})$$

$$\text{where} \quad r_j(F_{unknow}) = \|F_{unknow} - B\delta_j(\psi_{t,unknow})\|_2 \tag{10}$$

Their classification method has made great success in face recognition based on SRC, when the sample set is large, with carefully cropping and normalizing each sample offline for good performance. However, a large sample set and pretreatments of the samples are impossible for the efficiency requirement of object tracking.

Figs. 6 and 7 show samples from two classes and their corresponding sparse coefficient vectors. Fig. 6(a) and (b) are from the same object in [36] (redteam), and Fig. 6(c) and (d) are from the same object in [36] (egtest01). Fig. 7(a)–(d) show the sparse coefficients based on the same common sample set of Fig. 6(a)–(d), respectively. The values of L2-norm $\|\cdot\|_2$ between the four sparse coefficients are given in Fig. 7, where we can see that objects from the same class have similar sparse coefficient vectors (the smaller $\|\cdot\|_2$ values shown in red), indicating that the AdaSRA is feasible. Algorithm 2 below summarizes the multi-object tracking procedure.

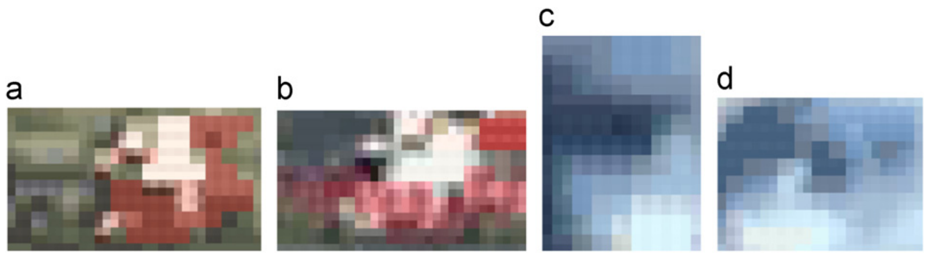
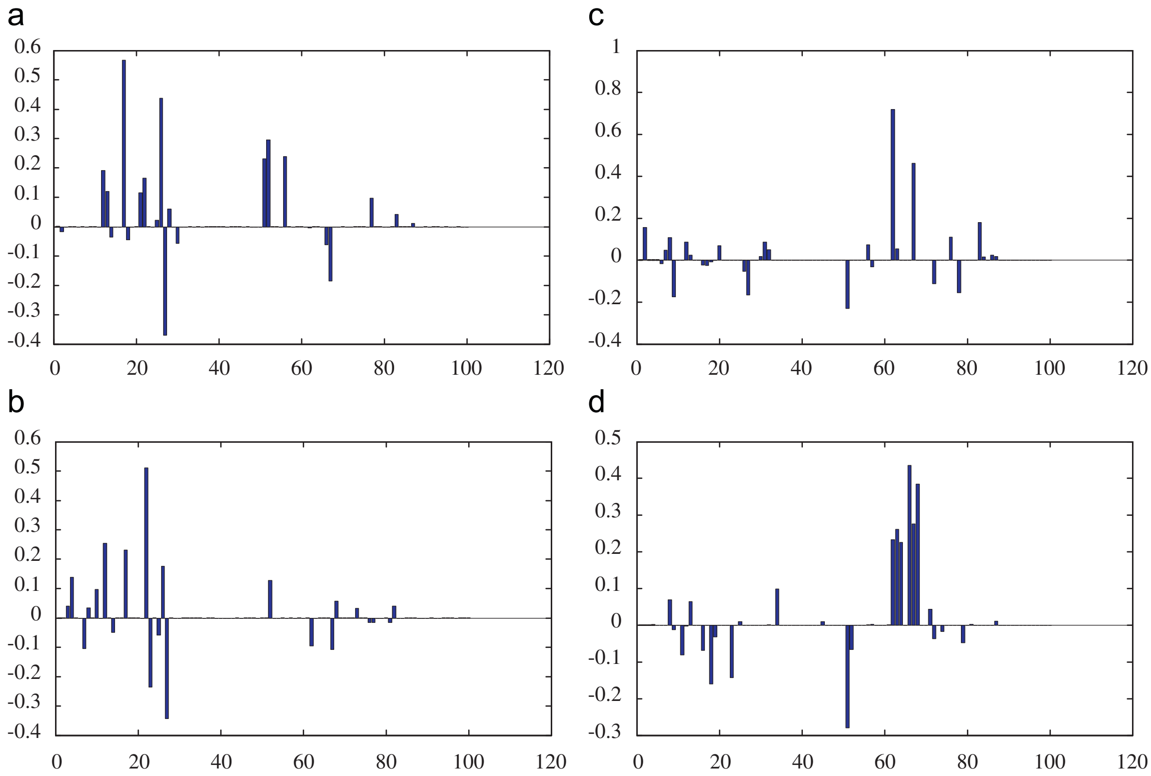


Fig. 6. Examples of the tracked objects during the multi-object tracking procedure: (a) and (b) are from the same object, and (c) and (d) are from another object.



$\|\psi_a - \psi_b\|_2 = 0.9859$ $\|\psi_a - \psi_c\|_2 = 1.4398$ $\|\psi_a - \psi_d\|_2 = 1.4670$

$\|\psi_b - \psi_c\|_2 = 1.2146$ $\|\psi_b - \psi_d\|_2 = 1.3799$ $\|\psi_c - \psi_d\|_2 = 0.9947$

Fig. 7. Illustration of sparse coefficients on multi-object tracking: (a)–(d) are the coefficient vectors of Figs. 6(a)–(d), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

```
Algorithm 2: Multi-object tracking via AdaSRA

1. Initialization ($t=0$).
   1.1 Initializing each tracked object;
   1.2 Constructing the common sample set $B$ for all objects;
   1.3 Calculating the reference sparse representation $\psi_{0,j}$ for each object.
2. Object Validation and Identification ($t>0$). In a new video frame:
   2.1 Predicting the prior position and reference sparse representation $\widetilde{\psi_{0,j}}$ of each tracked object in the new frame;
   2.2 Obtaining the instantaneous tracked result $F_{unknow}$ by a single object tracking procedure described in Algorithm 1;
```

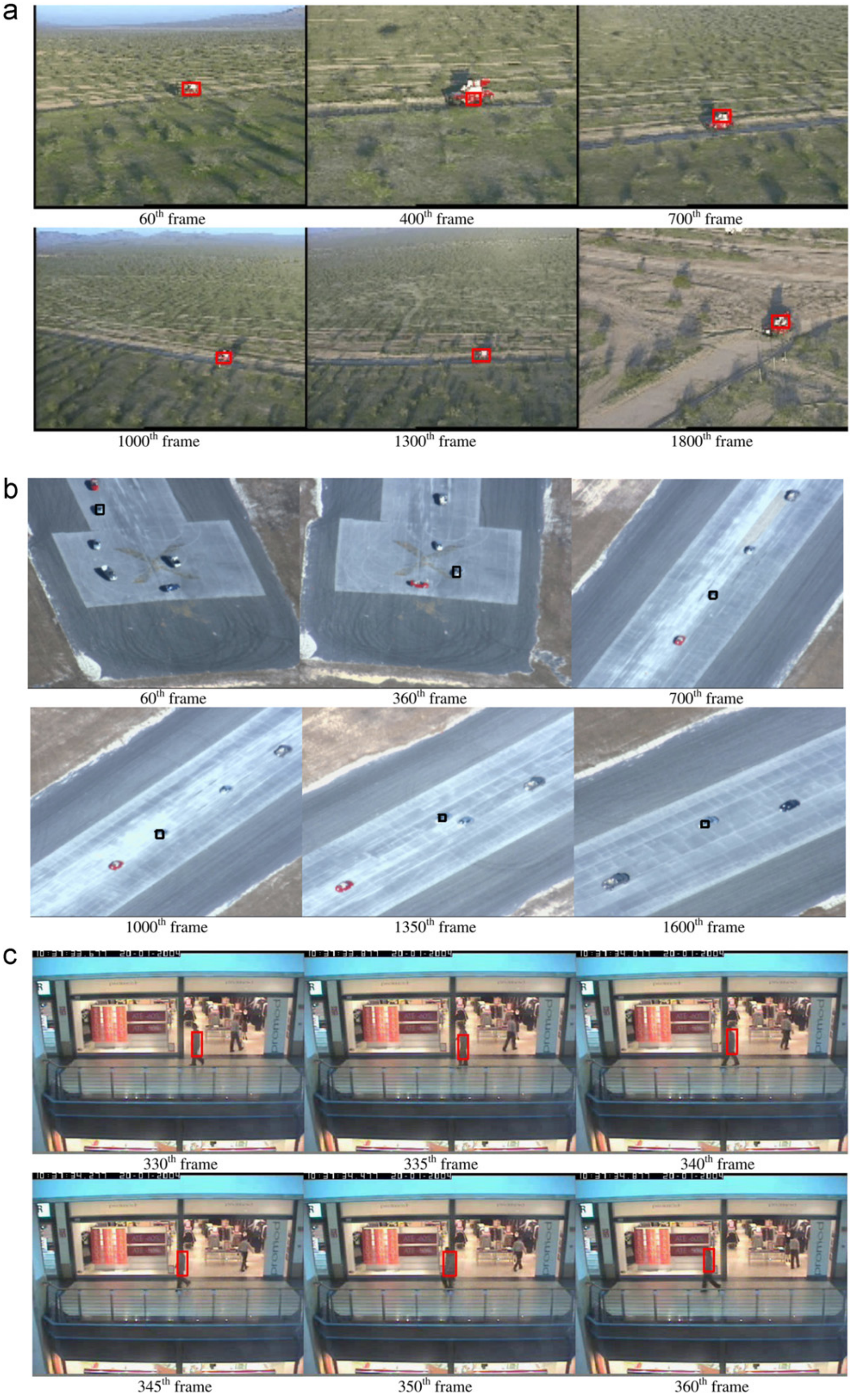


Fig. 8. Three tracking examples.

2.3 Calculating its instantaneous common sparse representation $\psi_{t,unknow}$ of the object based on the common sample set B;

2.4 Classifying the object with AdaSRA as in Eq. (9);

2.5 Correcting the posterior position and reference sparse representation $\psi_{0,j}$ of each object based on Kalman filter.

3. $\boldsymbol{t}=\boldsymbol{t}+\mathbf{1}$. If no updating, go to step 2 or end the tracking loop, otherwise go to step 4.

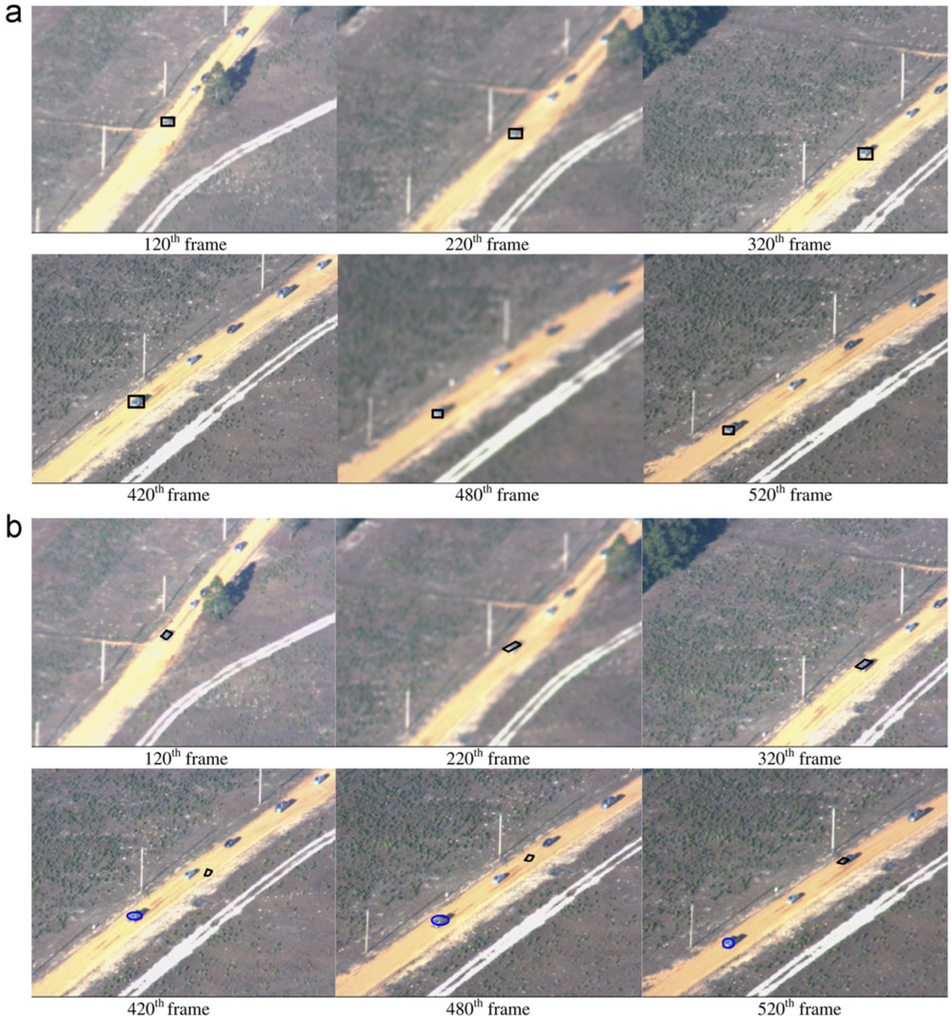


Fig. 9. Tracking results with deterioration in object images. The tracking result is marked with black rectangle and the ground truth is with blue ellipse, once there are tracking errors: (a) results of our proposed method and (b) results of the method in [26].

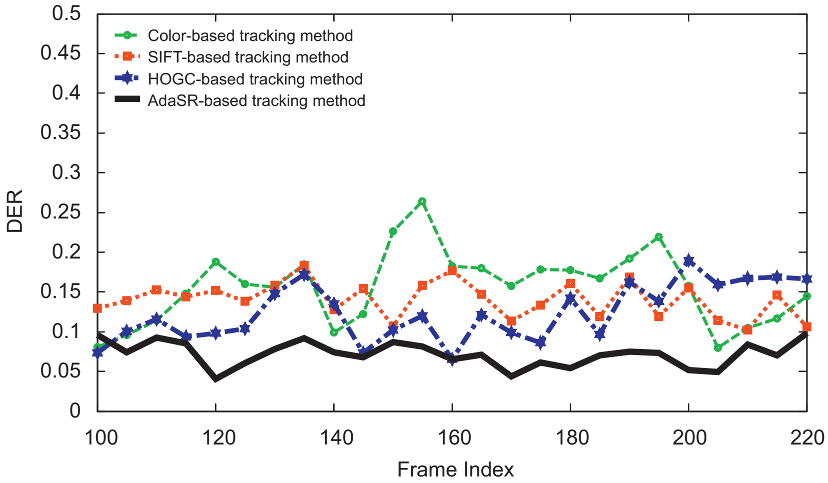


Fig. 10. Average DER of the four feature-based tracking methods.

4. **Common sample set updating ($t>0$).**
   4.1 Updating the sample set of each object;
   4.2 Go to step 1.2.

5. Experiments

In this section, experiments with comparisons are carried out to validate the proposed tracking approach. The experimental videos are selected from VIVID [36], CAVIAR [37] and our SDL data set [38]. The test videos consist of a variety of cases, including occlusions between tracking objects, lighting changes, scale variations, object rotations and complex backgrounds. Some of the videos are captured on moving platforms, and the target objects include moving pedestrians and vehicles. In the experiments, no image pre-processing is employed.

5.1. Validation of the Adaptive Sparse Representation (AdaSR)

There are various factors that make tracking challenging: different viewpoints (most of these sequences are captured by moving cameras), illumination changes, variations of the objects and backgrounds. Experiments on four challenging tracking examples are shown in Figs. 8 and 9 to illustrate the advantages of the proposed object representation.

The first test video in Fig. 8a is from the VIVID data set, and the object is a small jeep with a relative simple background and uniform object movement. The appearance of the object changes remarkably in size during the tracking process. The second video shown in Fig. 8b is from the VIVID data set. In this video, the car being tracked first loops around on a runway, then goes straight and speeds up. The car changes in both size and orientation remarkably during the tracking process. In addition, because of the similar color between the car and its background, the color features are not discriminative for tracking. The third test video in Fig. 8c is from the CAVIAR data set. In this video, the target is a person walking across the corridor. Since the person image is almost in gray, the color features are not informative for tracking. Furthermore, there are some mimic objects such as the columns, which are quite similar to the object both in color and in shape. The tracking conditions in Fig. 8(a)–(c) result in the degeneration of the objects representations in feature space, which always leads to the template drift and tracking failures.

Table 1
Video file list for average DER calculation.

Video test set	Video name
VIVID tracking video set	redteam egtest01 egtest02 egtest04
SDL tracking video set	xiangshan_ 0032 xiangshan_ 0043.
CARVIA tracking video set	Browse1 Fight_Chase OneStopMoveEnter1cor EnterExitCrossingPaths2front

AdaSR chooses a limited subset of representative samples as a basis for object representation, which is insensitive to object rotation, deterioration in object images, object's scale variation and complex background. What's more, the evolution of the AdaSR in filter framework can ensure the temporal consistency and adaptation of the appearances variations during the tracking process. The tracking results in Fig. 8 demonstrate that the proposed approach can effectively handle such tracking conditions.

In Fig. 9, we compare our proposed tracking method with the method in [26]. The test video is from the VIVID data set. In this video, the car being tracked goes straight along a road. During the tracking process, the object images have heavy deterioration (220th and 480th frames), which results in the degeneration of its representation in feature space and then may cause the template drift. We compare our proposed tracking method with the method in [26]. In Fig. 9a, our tracking method can track the object robustly, while the method in [26] loses the object in 420th, 490th and 520th frames shown in Fig. 9b.

To quantitatively evaluate the efficiency of the proposed AdaSR, we define a relative displacement error rate (DER) between the tracking results and the ground truth for performance evaluation.

$$\text{DER}=\frac{\text{Displacement error between tracked object position and groundtruth}}{\text{Size of the object}}$$

Firstly, we compare our proposed adaptive sparse representation with other three representative ones, including the color [14], SIFT [16] and HOGC [11] features. In the experiments, we use the

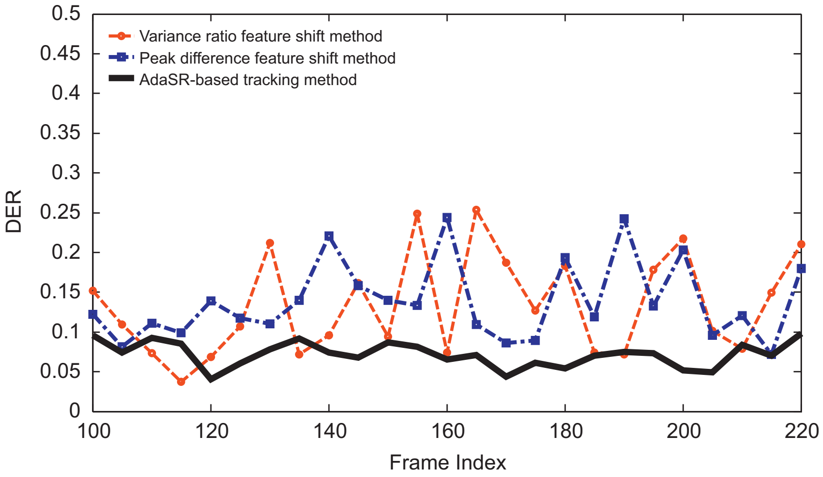


Fig. 11. Average DR of the three feature selection/evaluation-based tracking methods.

average DER of 10 video clips (listed in Table 1) to reflect the performance of each feature based tracking method. We use the same initialization for all the methods. The lower the average DER is, the better the tracking performance. The results of the four representations are shown in Fig. 10. It can be seen from the figure that the average DER of our AdaSR (about 0.04–0.1) is much smaller than those of the other three methods in almost the whole tracking process. These comparisons demonstrate that the

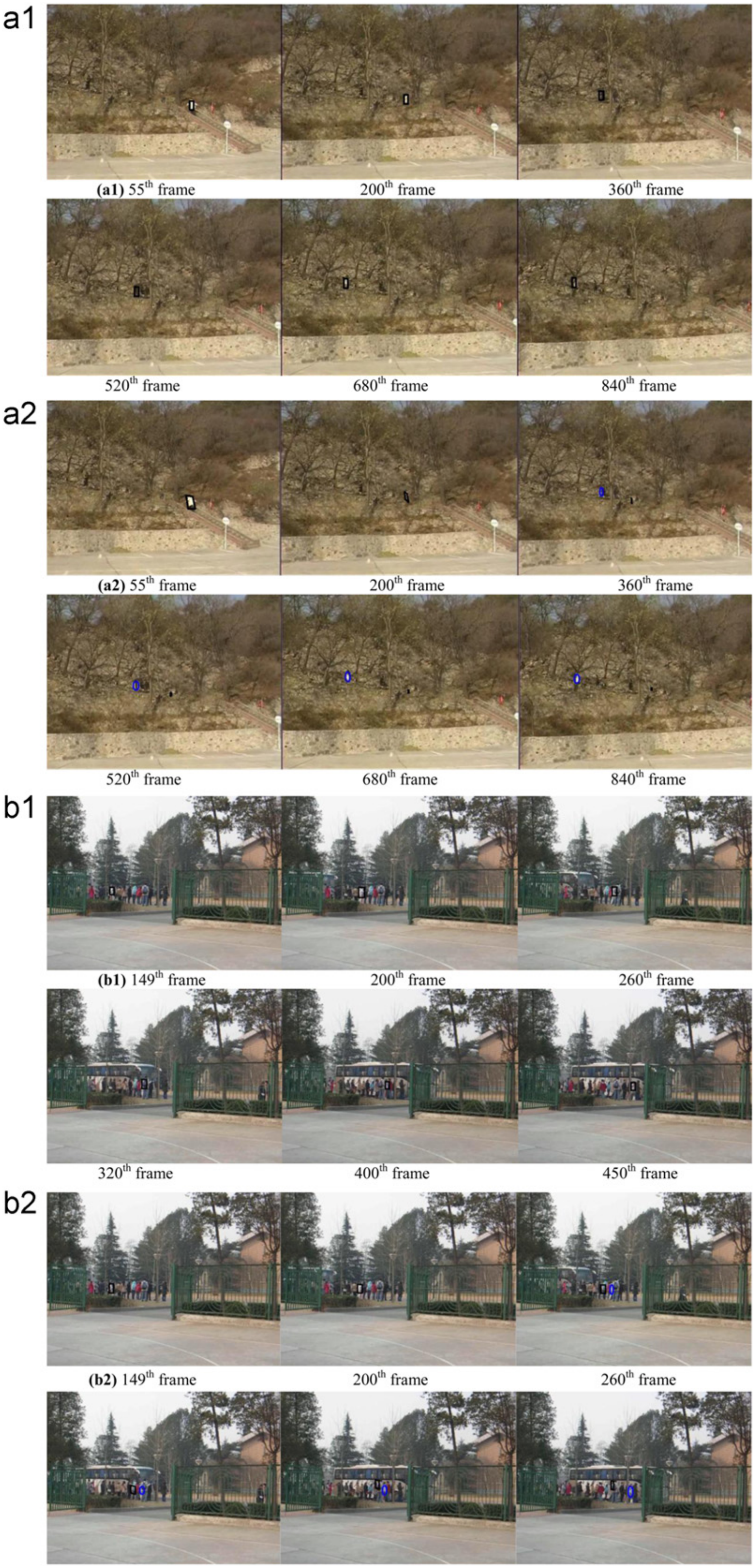


Fig. 12. Tracking results with partial occlusions and appearance variations of the tracked object. The tracking result is marked with black rectangle and the ground truth is with blue ellipse, once there are tracking errors: (a1) and (b1) results of our proposed method and (a2) and (b2) results of the method in [26].

proposed AdaSR has better performance in object representation than the others.

Furthermore, to demonstrate the adaptive efficiency of the proposed evolution of sparse representation in Kalman filter, we compare with two other representative feature selection/evaluation methods for adaptive visual tracking, including Variance ratio feature shift [13], and Peak difference feature shift tracking methods [13]. It can be seen in Fig. 11 that the average DER of our proposed approaches is much lower and holding a lower variation than those of the other methods, which indicates that the proposed object sparse representation with evaluation in filter frameworks has a better stability during the tracking process, which shows the better adaptation to object and background variation.

5.2. Tracking results under partial occlusions

Partial occlusion, which can easily and quickly change the appearances of the tracking objects, is another key factor that makes the tracking unstable. The first video shown in Fig. 12(a1) and (a2) is from the SDL data set. The main challenges of tracking in this video sequence arise from frequent partial occlusions of the object by other persons. When the object is occluded (260th and 400th frames), there are troubles using the initialized whole object template for tracking. By representing the object with a subset of similar samples, our approach succeeds in this example, since when a part of the object is occluded, we can track the object with other un-occluded parts of the object in the subset. The second video in Fig. 12(b1) and (b2) from the SDL data set is very challenging. There are not only serious occlusions (380th and 840th frames) but also appearance variations (560th frame) on the object. Our approach can also track the object correctly. The tracking results of these two videos show that the proposed approach can effectively deal with partial occlusions and appearance variations.

In the experiments, we still use the average DER (10 video clips listed in Table 1) to compare the performance between our proposed tracking method and other two representative ones, including Kalman Filter based tracking method [4] and Particle Filter based tracking method [5]. The results of the three methods are shown in Fig. 13. It can be seen that the average DER of our proposed approaches (about 0.04–0.1) is much lower than those of the other methods, which validate that the proposed object tracking can better handle the partial occlusions compared with the traditional ones.

5.3. Tracking results of multiple objects

In the experiments of multi-object tracking, there are three classes totally (three tracked objects). We select a test object, which belongs to class one, for object classification experiment.

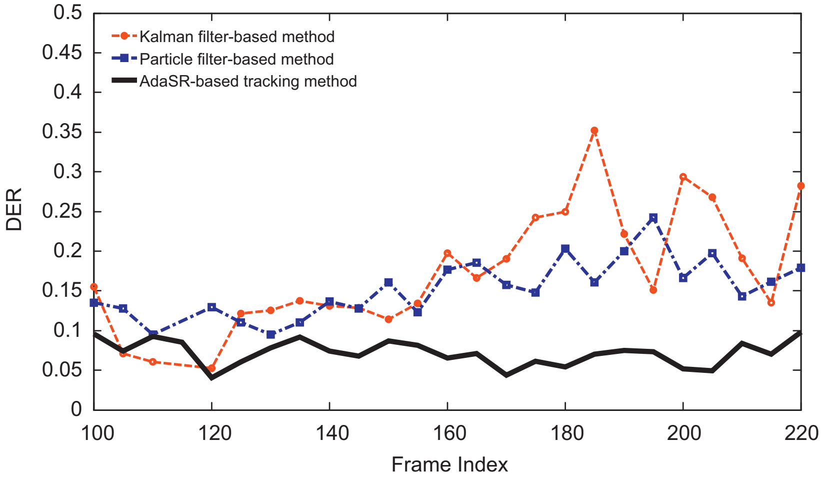


Fig. 13. Average DER of three tracking methods under partial occlusions.

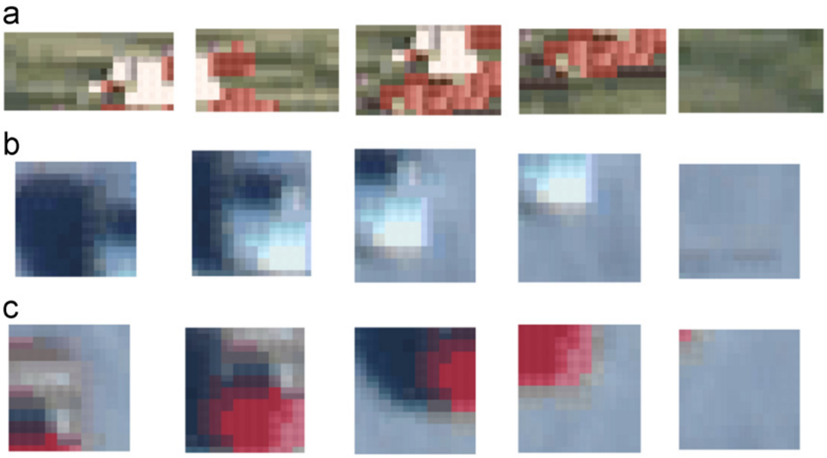


Fig. 14. Selected examples in the common sample set: (a) samples from class 1, (b) samples from class 2, and (c) samples from class 3.

Some samples from each class in the common sample set are shown in Fig. 14. We test the classification performance of AdaSRA based on Eq. (9) and SRC based on Eq. (10).

In experiment one, we construct the common sample set with 50 samples from each class, and the results are shown in Fig. 15b. We classify the object to the class with the smallest $r_j(F_t)$ in both AdaSRA and SRC. It can be seen from the figure that SRC misclassifies the test object into class 2, according to Fig. 15b (SRC), while our approach correctly classify the object to class 1, according to Fig. 15b (AdaSRA). When the sample set of each class is expanded to 100 samples in experiment two, the classification results of the both methods are given in Fig. 15c. The proposed approach can obviously classify the object shown in Fig. 15c (AdaSRA). SRC makes some improvements, but still has misclassification as shown in Fig. 15c (SRC).

The following tracking results in Fig. 16 are obtained by our proposed multi-object tracking approach based on AdaSRA. Fig. 16a shows the tracking results of two cars, which rotate around on a runway (330th and 860th frames) and intersect with other similar vehicles (510th and 690th frames). The results show that our multi-object tracking method can correctly track and classify each object. Fig. 16b demonstrates the tracking results of the four objects.

6. Conclusions

Object representation is very important to improve the adaptability of visual object tracking. In this paper, we have proposed a novel object tracking approach based on adaptive sparse representation by exploiting the L1-norm minimization in sample space and Kalman filter and then extended the approach to multi-object tracking by analyzing the adaptive reference sparse representation based on the common sample set. The tracking results with comparisons to other representative methods are provided, which indicates that the proposed tracking approach achieves state-of-the-art performance, even under partial occlusion, object distortion and appearance variations of both objects and their backgrounds. The experimental results of multi-object tracking demonstrate the effectiveness of AdaSRA on classifying multiple objects.

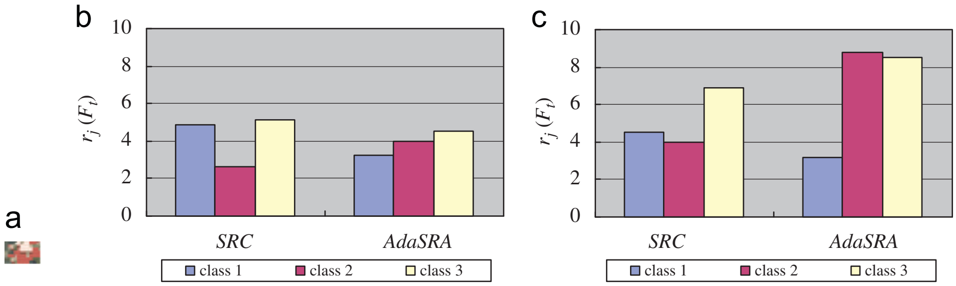


Fig. 15. Illustration of the classification results: (a) a test object from class 1, (b) the results of SRC in [22] and AdaSRA based on 50 samples for each class, and (c) the results of SRC in [22] and AdaSRA based on 100 samples for each class.

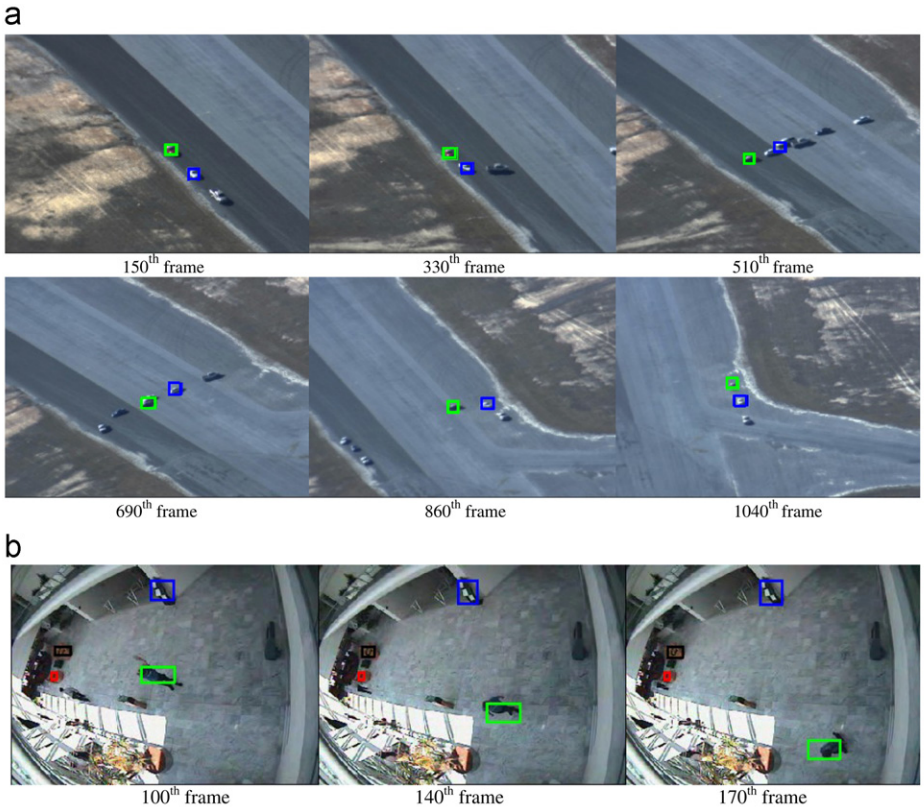


Fig. 16. The results of multi-object tracking.

The new concepts and techniques introduced in this paper include the sample set, object representation based on adaptive sparse representation, and object classification based on AdaSRA. And in the evaluation process, we extend the function of Kalman filter for evolution of sparse representation, which is novel in both visual object tracking and L1-norm minimization researches.

A known issue in the proposed tracking method is that the whole object occlusion problem has not been solved yet, for our proposed approach cannot predict the position of the object in the following video frames. This issue should be considered in the future work.

Aknowledgement

The authors would like to thank Guorong Li and the anonymous reviewers for their constructive comments. This work is supported in part by National Basic Research Program of China (973 Program) with nos. 2011CB706900 and 2010CB731800, and National Science Foundation of China with nos. 61039003 and 60872143.

References

[1] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Proceedings of the IEEE Conference on CVPR, 1999, pp. 246–252.
[2] G. Bradski, Real time face and object tracking as a component of a perceptual user interface, in: Proceedings of IEEE Workshop on Applications of Computer Vision, 1998, pp. 214–219.
[3] N. Papanikolopoulos, P. Khosla, T. Kanade, Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision, IEEE Transactions on Robotics and Automation 9 (1993) 14–35.
[4] E. Cuevas, D. Zaldivar, R. Rojas, Kalman filter for vision tracking, Technical Report B, Fachbereich Mathematikund Informatik, Freie Universität Berlin, 2005.
[5] M. Isard, A. Blake, Condensation-conditional density propagation for visual tracking, IJCV (1998) 5–28.
[6] S. Baker, and I. Matthews, Lucas-kanade 20 years on: a unifying framework, IJCV (2004) 221–255.
[7] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proceedings of the IEEE Conference on CVPR, 2000, pp. 142–149.
[8] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on CVPR, 1994, pp. 593–600.
[9] J. Wang, X. Chen, W. Gao, Online selecting discriminative tracking features using Particle Filter, in: Proceedings of the IEEE Conference on CVPR, 2005, pp. 1037–1042.
[10] J.Q. Wang, Y.S. Yagi, Integrating color and shape-texture features for adaptive real-time object Tracking, IEEE Transactions on Image Processing 17 (2008) 235–240.
[11] Z.j. Han, Q.x. Ye, J.b. Jiao, Online feature evaluation for object tracking using Kalman Filter, in: 19th International Conference on Pattern Recognition, 2008.
[12] Z.j. Han, Q.x. Ye, J.b. Jiao, Feature evaluation by particle filter for adaptive object tracking, in: Proceedings of the SPIE Visual Communication and Image Processing, 2009.
[13] R. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005).
[14] T.D. Grove, K.D. Baker, T.N. Tan, Colour based object tracking, in: 14th International Conference on Pattern Recognition, vol. 2, 1998.
[15] A. Yilmaz, X. Li, B. Shah, Contour based object tracking with occlusion handling in video acquired using mobile cameras, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1531–1536.
[16] Y.J. Li, J.F. Yang, R.B. Wu, F.X. Gong, Efficient object tracking based on local invariant features, in: IEEE International Symposium on Communications and Information Technologies, 2006, pp. 697–700.
[17] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Proceedings of the IEEE Conference on CVPR, 2006, pp.798–805.
[18] F.L. Wang, S.Y. Yu, J. Yang, A novel fragments-based tracking algorithm using mean shift, in: 19th International Conference on Control, Automation, Robotics and Vision, 2008, pp. 694–698.
[19] Y.Wu, J.Q. Wang, H.Q. Lu, Robust Bayesian tracking on Riemannian manifolds via fragments-based representation, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 765–768.
[20] D. Chen, J. Yang, Robust object tracking via online spatial bias appearance model learning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2007) 2157–2169.
[21] T. Serre, Learning a dictionary of shape-components in visual cortex: comparison with neurons, humans and machines, Ph.D. Dissertation, MIT, 2006.
[22] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008).
[23] J. Rissanen, Modeling by shortest data description, Automatica (1978) 465–471.
[24] M. Hansen, B. Yu, Model selection and the minimum description length principle, Journal of the American Statistical Association (2001) 746–774.
[25] J. Yang, A.F. Frangi, J.Y. Yang, D. Zhang, Z. Jin, KPCA Plus LDA: a complete Kernel Fisher discriminant framework for feature extraction and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2005) 230–244.
[26] X. Mei, H. b. Ling, Robust visual tracking using L1 minimization, in: IEEE Conference on ICCV, 2009.
[27] A.G. Daronkolaei, S. Shiry, M.B., Menhaj, Multiple targets tracking for mobile robots using the JPDAF algorithm, in: IEEE Conference on TAI, 2007, pp. 137–145.
[28] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 810–815.
[29] W.C. Huang, T.C. Hwann, A square-root sampling approach to fast histogram-based search, in: Proceedings of the IEEE Conference on CVPR, 2010.
[30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on CVPR, 2005, pp. 1063–6919.
[31] E. Canđes, Compressive sampling, The International Congress of Mathematicians, 2006.
[32] R. Tibshirani, Regression shrinkage and selection via the LASSO, Journal of the Royal Statistical Society B (1996) 267–288.
[33] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Review (2001) 129–159.
[34] D. Donoho, Y. Tsaig, Fast solution of L1-norm minimization problems when the solution may be sparse, http://www.stanford.edu/tsaig/research.html, 2006.
[35] R.G. Brown, P.Y.C. Hwang, Introduction to Random Signals And Applied Kalman Filtering [M], John Wiley & Sons, Inc., New York, 1992.
[36] VIVID tracking evaluation web site at: 〈http://www.vividevaluation.ri.cmu.edu/datasets/datasets.html〉.
[37] CAVIAR test case scenarios at: 〈http://homepages.inf.ed.ac.uk/rbf/CAVIAR〉.
[38] SDL data set at: 〈http://coe.gucas.ac.cn/SDL-HomePage/〉.

Zhenjun Han received his B.S. degree in software engineering from Tianjin University (TJU), Tianjin, in 2006. Since 2009, he has been a Ph.D. candidate of the Graduate University of Chinese Academy of Sciences, Beijing, China. His research interests include image processing, intelligent surveillance, etc.

Jianbin Jiao received the B.S., M.S. and Ph.D. degrees in mechanical and electronic engineering from Harbin Institute of Technology of China (HIT), Harbin, in 1989, 1992, and 1995, respectively. From 1997 to 2005, he was an associate professor of HIT. Since 2006, he has been a professor of the Graduate University of Chinese Academy of Sciences, Beijing. His research interests include image processing, pattern recognition, intelligent surveillance, etc.

Baochang Zhang received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 1999, 2001, and 2006, respectively. From 2006 to 2008, he was a Research Fellow with the Chinese University of Hong Kong and Griffith University, Australia. Currently, he is a lecturer with Beihang University, China. His research interests include pattern recognition, machine learning, face recognition, and wavelets.

Qixiang Ye received his B.S. and M.S. degrees in mechanical & electronic engineering from Harbin Institute of Technology of China (HIT), Harbin, in 1999 and in 2001, respectively. He received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2006. Since 2009, he has been an associate professor at the Graduate University of the Chinese Academy of Sciences, Beijing. His research interests include image processing, pattern recognition, statistic learning, etc.

Jianzhuang Liu (M'02–SM'02) received the B.E. degree from Nanjing University of Posts and Telecommunications, China, in 1983, the M.E. degree from Beijing University of Posts and Telecommunications, China, in 1987, and the Ph.D. degree from The Chinese University of Hong Kong in 1997. From 1987 to 1994, he was a faculty member with the Department of Electronic Engineering, Xidian University, China. From August 1998 to August 2000, he was a research fellow at the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore. Then he was a postdoctoral fellow with the Chinese University of Hong Kong for several years. He is now an Assistant Professor in the Department of Information Engineering, The Chinese University of Hong Kong. His research interests include image processing, computer vision, machine learning, and graphics.