Machine Vision and Applications
DOI 10.1007/s00138-014-0632-3

ORIGINAL PAPER

Multi-scale patch-based sparse appearance model for robust object tracking

Chengjun Xie · Jieqing Tan · Peng Chen · Jie Zhang · Lei He

Received: 20 September 2013 / Revised: 26 May 2014 / Accepted: 6 July 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract When objects undergo large pose change, illumination variation or partial occlusion, most existing visual tracking algorithms tend to drift away from targets and even fail to track them. To address the issue, in this paper we propose a multi-scale patch-based appearance model with sparse representation and provide an efficient scheme involving the collaboration between multi-scale patches encoded by sparse coefficients. The key idea of our method is to model the appearance of an object by different scale patches, which are represented by sparse coefficients with different scale dictionaries. The model exploits both partial and spatial information of targets based on multi-scale patches. Afterwards, a similarity score of one candidate target is input into a particle filter framework to estimate the target state sequentially over time in visual tracking. Additionally, to decrease the visual drift caused by frequently updating model, we present a novel two-step object tracking method which exploits both the ground truth information of the target labeled in the first frame and the target obtained online with the multi-scale patch information. Experiments on some publicly available benchmarks of video sequences showed that the similarity involving complementary information can locate targets more accurately and the proposed tracker is more robust and effective than others.

Keywords Object tracking · Multi-scale patch · Sparse representation · Appearance model

C. Xie · J. Tan · L. He
School of Computer and Information, Hefei University of Technology, Hefei 230009, China
e-mail: cjxie@iim.ac.cn

C. Xie · J. Zhang
Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China

P. Chen (✉)
Institute of Health Sciences, Anhui University, Hefei 230601, Anhui, China
e-mail: bigeagle@mail.ustc.edu.cn

1 Introduction

Object tracking is an important problem and plays a crucial role in many practical applications such as video surveillance, human motion understanding, interactive video processing and so on. Although many trackers have been proposed and have made successes under various scenarios, it is still challenging in object tracking because the appearance of an object may be changed drastically while undergoing significant pose change, illumination variation or partial occlusion. Such a thorough review can be found in [1], which presented a typical tracking system consisting of three components: an appearance model, which evaluates the similarity of the object of interest being at different particular locations; a motion model, which locates the target over time; and a search strategy for finding out the most likely location of the target in the current frame. In this paper, we focus on the design of a robust appearance model and a two-step tracking strategy with particle filtering.

Recent years have seen a significant progress in effective appearance model for robust object tracking. One successful approach was a class of appearance modeling techniques named sparse representation [2–4]. Several tracking methods based on sparse representation have been proposed [5–8]. In these cases, tracking problems were formulated to find a sparse approximation using template subspace, and further experiments showed that sparse representation was efficient and adaptable to address the aforementioned chal-

Published online: 27 August 2014

lenges, especially for partial occlusion. However, besides the high computational cost of the tracking problem, another drawback is the limitation of the appearance model. Since the templates directly cropped from target images are very limited, all the above trackers may fail due to the inaccurate linear representation for the target. To alleviate the problem, local patch-based sparse representation model was introduced [9,10]. Liu et al. [9] proposed a tracking method which employs histograms of sparse coefficients and mean-shift algorithm for object tracking; however, it tracked target patches only with a static local dictionary and failed in dynamic scenes. Jia et al. [10] adopted an alignment pooling method to scan across local patches based on sparse coefficients. Although these local patch-based trackers have demonstrated good robustness in many videos, they are patch size sensitive and their optimal patch sizes were varied with target template size.

Inspired by the works mentioned above, to make the patch-based trackers robust, a multi-scale patch tracking algorithm involving information from different scales is proposed based on local sparse representation. Since target appearance exhibits distinct spatial structures and characteristics on different scales, combining the spatial information and structural information on different scales could not only lead to much tracking improvement, but also provide an effective appearance model for robust tracking. First, the proposed method samples local image patches with different patch size within the target region and further constructs a multi-scale local dictionary. Then, similarly to [10], the single-scale similarity measure score of one candidate region is obtained across the local patches by alignment-pooling method. Finally, the multi-scale similarity measure of one candidate region is computed by applying different scale weights to the corresponding single-scale similarity.

Although the proposed method is adapted by updating the target appearance model with respect to new input targets to be tracked, the main issue is that observation noise is inevitably used for the update and, correspondingly, the target template is changed frequently with new tracking results, thereby causing drift. To avoid significant drift, similarly to [11,12], a novel two-step tracking strategy is proposed by use of multi-scale patch information. In the first step, we use a dynamical appearance model and therefore define an dynamical likelihood function to estimate the target state within particle filtering framework. In the second step, since the ground truth plays a key role in determining whether a new tracking result is effective during the tracking process and it is only available in first frame, the target labeled in the first frame is used as a static appearance model with multi-scale patch information and therefore defines a static likelihood function to select the best accurate target position resulting from the first step.

The first contribution of this paper is the presentation of the dynamical appearance model that integrates multi-scale patches by sparse coefficients. The appearance model exploits both partial information and spatial information of the target based on patches on multi-scale level. The second contribution is the proposal of a novel multi-scale similarity measure on each patch of one candidate region. The last contribution is a novel two-step particle filtering method. The proposed two-step approach combines static appearance model with dynamical appearance model, thereby alleviating the drift problem when updating the appearance model.

2 Related works

Many works have focused on constructing target appearance model, which is a key part of object tracking. An effective object representation should have a strong description or discrimination power to distinguish targets from background. In general, most of the tracking algorithms can be categorized as either generative [13–22] or discriminative [23–27] based on their appearance models. Color histogram was one of the most widely used appearance models [18] in many tracking algorithms [13,14,28]. However, those trackers did not work well when objects underwent illumination change and/or large pose change. To address these issues, Ross et al. developed an online subspace learning model to account for appearance variation [15]. Furthermore, Kwon et al. [16] tracked the target successfully via visual tracking decomposition (VTD) with multiple and dynamic observation models. In [17], Kwon et al. extended the VTD method and proposed a visual tracker sampler framework that tracked a target by searching for the appropriate tracker in each frame. However, a main drawback of them is the limitation of the appearance methods to model holistic object appearance within a generative framework.

Discriminative methods have shown that training a model via a discriminative classifier often performed well in discriminating objects from the background [23,25,26]. Avidan [25] trained a support vector machine (SVM) classifier offline and applied its extension in an optimal flow framework for object tracking. Furthermore, Avidan et al. [23] developed an online boosting method for tracking targets, which was an ensemble tracker that constructed a strong classifier composed of a set of weak classifiers. Grabber et al. [26] utilized online AdaBoost algorithm with a proposed novel feature selection method. Moreover, Parag et al. [27] applied boosting method in object tracking, but its classifier-updating method consists of a set of weak classifiers that are updated with the change of the background. In contrast with them, Babenko et al. [29] used multiple instance learning (MIL) instead of traditional supervised learning to handle ambiguous binary data obtained online. However, a major challenge

in discriminative methods is how to choose positive and negative samples when updating the dynamical appearance model. Moreover, most of discriminative trackers took the current object location as one positive sample and sampled its neighborhoods for negatives, which might degrade the appearance model and cause drift when the current object location is imprecise.

Recently, a sparse representation framework in [2] presented a novel path for solving the problem of object occlusion and has been successfully applied in robust face recognition [30]. Motivated by the work, many methods adopted sparse representation model for tracking objects [5–10,31]. In [5], each target candidate was represented as a linear combination of a set of updated templates online consisting of target templates and trivial templates. The candidate with the smallest error to target template reconstruction was regarded as the tracking outcome. Moreover, Bai et al. [8] presented a structured sparse appearance model, which can reduce the computational cost, for tracking objects, where block orthogonal matching pursuit was adopted to solve the structured sparse representation problem. Liu et al. [9] modeled target as histograms of local sparse representation and integrated them into mean-shift tracking framework for object tracking. More recently, in [10], a tracking algorithm based on alignment pooling method with sparse coefficients was proposed. Besides the high computational cost of the tracking process, another drawback of those trackers is the limitation of the appearance model to model object appearance as single-scale patch.

In this paper, we propose a novel object representation method combining multi-scale patch and sparse representation. The object representation, whose appearance model is composed of multi-scale patch features with the corresponding sparse coefficients, provides a more flexible mechanism to deal with the problem of appearance change. Furthermore, motivated by the work [11,12,32], a two-step tracking strategy is proposed to reduce drift.

3 Multi-scale patch-based sparse representation model

3.1 Sparse representations

Sparse representations have attracted a great deal of attention in signal processing and have been widely used in many fields such as visual tracking [5,7,8,31]. Consider a signal $y \in \mathbb{R}^n$, which can be represented as a linear representation of basic elements from a dictionary $D \in \mathbb{R}^{n\times c}$ that is composed of atoms $\{d_M\}_{M=1}^{c}$. A representation of the signal y based on the dictionary D is any vector $x \in \mathbb{R}^c$ that satisfies:

$$y = Dx + z, \tag{1}$$

where the dictionary D is said to be over-complete if $n < c$ and z is a noise term with bounded energy $\|z\|_2 < \varepsilon$. However, the solution of x is generally non-sparse with many nonzero elements. To obtain a linear combination of only a few elements to approximate the signal y, the problem can be formally described by

$$\hat{x_0} = \arg\min \|x\|_0 \ \text{ subject to } \ \|y - Dx\|_2 < \varepsilon, \tag{2}$$

where $\|\cdot\|_0$ is the l_0 norm which counts the number of nonzero elements, $\|\cdot\|_2$ is the l_2 norm, and the parameter ε denotes the level of reconstruction error. Since the combinatorial l_0-norm minimization is an NP-hard problem, l_1-norm minimization is applied and thus formulated as

$$\hat{x_1} = \arg\min \|x\|_1 \ \text{ subject to } \ \|y - Dx\|_2 < \varepsilon. \tag{3}$$

3.2 Patch-based local sparse appearance model

In this work, a local sparse representation is used to model the appearance of target patches and a set of sparse coefficients are collected to represent them. Given a set of target templates $X = \{X_i | i = 1 : n\}$, a set of image patches $D = \{d_j | j = 1 : n \times K\}$ inside the target region is obtained by sliding a window with fixed size to sample image patches, where $d_j \in \mathbb{R}^d$ is the j-th column representing a vectorized image patch, d is the dimensionality of image patches, n is the number of target templates and K is the number of local image patches from the target region. Due to containing overlapped image patches, the over-complete dictionary is constructed by D.

Let $P = \{p_i | i = 1 : K\}$ denote the vectorized image patches extracted from a target candidate, where $p_i \in \mathbb{R}^d$ is the ith local image patch. With the dictionary D, each p_i will have a corresponding vector with reconstruction coefficients $\alpha_i \in \mathbb{R}^{(n\times K)\times 1}$, which is computed by:

$$\hat{\alpha_i} = \arg\min \|\alpha_i\|_1 \ \text{ subject to } \ \|p_i - D\alpha_i\|_2 < \varepsilon. \tag{4}$$

The sparse codes $A = [\hat{a}_1, \hat{a}_2, ..., \hat{a}_K]$ of all the image patches in one candidate are calculated to represent the candidate. To represent local patch i at a certain position of the candidate, the sparse coefficients of the ith patch are divided into n segments (see Fig. 1), i.e., $\hat{a}_i^{\mathrm{T}} = [\hat{a}_{i1}^{\mathrm{T}}, \hat{a}_{i2}^{\mathrm{T}}, ..., \hat{a}_{in}^{\mathrm{T}}]$, according to the dictionary $D = \{d_j | j = 1 : n \times K\}$ obtained by target templates $X = \{X_i | i = 1 : n\}$, where $\hat{a}_{im} \in \mathbb{R}^{K\times 1}$ denotes the mth segment of the coefficients $\hat{a}_i$. Then these $\hat{a}_{im}$ are weighted by:

$$Z_i = \sum_{m=1}^{n} \hat{a}_{im}, \quad i = 1, 2, ..., K, \tag{5}$$

where vector Z_i corresponds to the ith local patch. After obtaining Z_i, each local patch at a certain position of one candidate is represented by sparse codes at different positions of

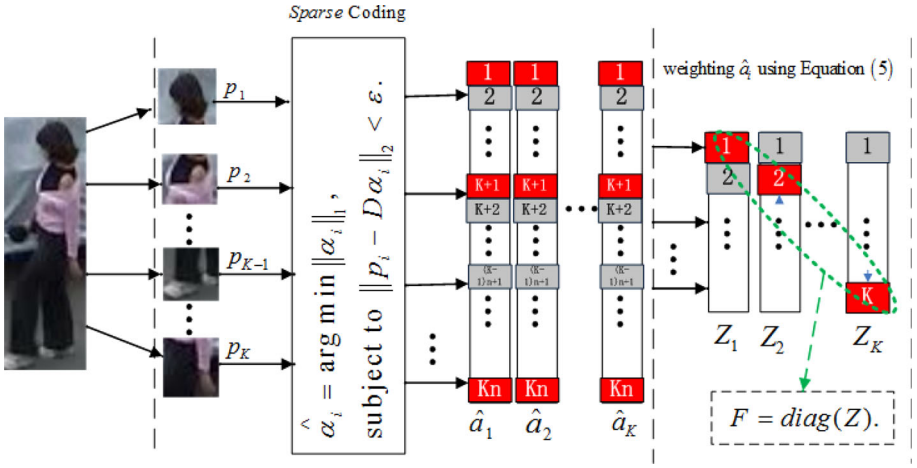


Fig. 1 Flowchart of patch-based local sparse appearance model

the dictionary D. The local appearance variation of a patch can be described by the sparse codes at the same positions of the dictionary D. Therefore, in [10] an alignment-pooling method was proposed by taking the diagonal elements of the square matrix Z as pooled feature, i.e., $F = \text{diag}(Z)$, where F represents the vector composed of the pooled features of target candidate and indicates local appearance variation between target and templates based on the locations of structural image patch by the dictionary D. The flowchart of patch-based local sparse appearance model is shown in Fig. 1.

3.3 Multi-scale patch appearance model by sparse representation

Although the proposed patch-based local sparse appearance model with alignment-pooling process can capture both partial information and spatial information, patch scale, or called patch size, will greatly influence the tracking performance. To get rid of the impact of patch size, the problem is solved by fusing the pooled features of multi-scale patch adaptively, and therefore it can not only be free of the scale selection problem but also exploit the complementary partial and spatial information across different scales to improve tracking results.

In our algorithm, we model the appearance of target patches with different patch sizes and the corresponding sparse coefficients are collected to represent target patches. A set of overlapped local image patches inside the target region with a spatial layout is obtained by sliding window with different sizes to sample image patches. A sample rectangle inside the target region is specified by $R = (w, h, s, r, \alpha)$, where w and h are the width of the target image and the height of the target image, respectively, s denotes scale, r is patch scale or called patch size and $0° \leq \alpha \leq 360°$. These local patches are used for dictionary $D^s = \{d_j^s | j = 1 : n \times K, s = 1, 2, ..., L, r = 2 \times s + 2\}$, where $d_j^s \in \mathbb{R}^{d^s}$ is the jth column for representing a vectorized image patch, d^s is the dimensionality of image patches, K is the number of local image patches under scale s and L is the number of different scales (L is set to 7 in this work), Seven scales with patch sizes 4×4, 6×6, 8×8, 10×10, 12×12, 14×14 and 16×16 are used here.

Let $P^s = \{p_i^s | i = 1 : K, s = 1, 2, ..., L.\}$ be the vectorized image patches extracted from a target candidate under different patch scales, where p_i^s is the ith local image patch under patch scale s. With the dictionary D^s, each p_i^s will have a corresponding vector composed of reconstruction coefficients $\alpha_i^s \in \mathbb{R}^{(n \times K) \times 1}$, which is computed by:

$$\hat{a}_i^s = \arg\min \|\alpha_i^s\|_1 \text{ subject to } \|p_i^s - D^s \alpha_i^s\|_2 < \varepsilon. \quad (6)$$

When the sparse codes $B^s = [\hat{a}_1^s, \hat{a}_2^s, ..., \hat{a}_K^s]$ of one candidate are computed under different patch scales, they are used to represent the candidate. To represent each local patch at a certain position of the candidate, the pooled feature is redefined as:

$$Z_i^s = \omega_s \sum_{m=1}^{n} \hat{a}_{im}^s \quad , i = 1, 2, ..., K_s, \quad (7)$$

where $\hat{a}_{im}^s \in \mathbb{R}^{K \times 1}$ denotes the mth segment of the coefficients $\hat{a}_i^s$, $C$is a normalization term and vector Z_i^s corresponds to the i-th local patch on scale s. All the Z_i^s of local patches form a square matrix Z^s and then the vector of pooled feature can be obtained by $F^s = \text{diag}(Z^s)$, $s = 1, 2, ..., L$ under scale s. Moreover, ω_s is the weight of pooled features and indicates that the larger the weight the more important the pooled features are. When the tracked objects undergo appearance deformation or partial occlusion, the image patches that are not occluded or deformed can still be represented with small reconstruction error, whereas the occluded or deformed patches have large reconstruction error. In this paper we develop a strategy to assign weights for different scales adaptively, by reconstruction errors using sparse representation. Let O^s denote a descriptor of deformation of the corresponding local patch under scale s. It is defined by

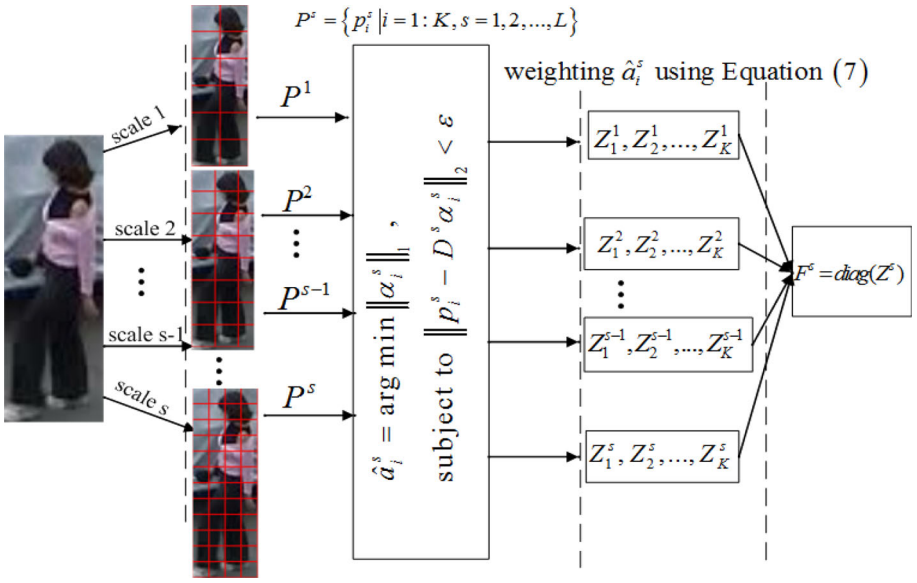


Fig. 2 Flowchart of multi-scale patch-based appearance model

$O_i^s = \begin{cases} 1 & e_i^s \geq e_0 \\ 0 & \text{otherwise} \end{cases}$, where $e_i^s = \|p_i^s - D^s \hat{a}_i^s\|_2^2$ is the reconstruction error of the local patch p_i^s under patch scale s and e_0 is a threshold which indicates whether the patch in one candidate target is deformed or not. The weight of pooled features under scale s is defined by:

$$\omega_s = 1 - \beta^s, \ \beta^s = \sum_{i=1}^{K} O_i^s / K, \tag{8}$$

where ω_s is the weight associated with the pooled features of different scales, K is the number of local image patches and $\sum_{i=1}^{K} O_i^s$ is the number of local image patches that are occluded or deformed for one candidate under scale s and β^s denotes the deformation ratio of the target under different scales. In this case, the larger the value of β^s is, the more heavy the target will be deformed or occluded and the smaller the weight ω_s is thereby assigned for combination. The flowchart of multi-scale patch-based appearance model is shown as Fig. 2.

4 Two-step object tracking method with particle filtering

4.1 Particle filtering

Particle filter [33,34] provided a convenient framework for estimating and propagating the posterior probability density functions of state variables. In this paper, to form a robust tracking algorithm, a multi-scale patch similarity is embedded into the particle filter framework. Given observations of the target up to time t, $y_{1:t} = \{y_1, ..., y_t\}$, the current target state s_t can be estimated by maximizing a posterior (MAP) that associates with the highest likelihood:

$$s_t = \arg\max_{s_t} p(s_t \mid y_{1:t}), \tag{9}$$

where $p(s_t|y_{1:t})$ is the posterior probability and is recursively computed as

$$p(s_t \mid y_{1:t}) \propto p(y_t \mid s_t) \int_{s_{t-1}} p(s_t \mid s_{t-1}) \, p(s_{t-1} \mid y_{1:t-1}) \, ds_{t-1}, \tag{10}$$

where $p(y_t|s_t)$ is the observation model or likelihood function that estimates the likelihood of the state s_t, given an observation y_t and $p(s_t|s_{t-1})$ is the motion model that predicts the current state given the previous state.

Here, similar to [5], an affine image warping is applied to model target motion of two consecutive frames. Let $s_t = (l_1, l_2, \mu_1, \mu_2, \mu_3, \mu_4)$ be the parameter vector for affine transformation, where $\mu_1, \mu_2, \mu_3, \mu_4$ are the deformation parameters which represent rotation angle, scale, aspect ratio and skew direction at time t, respectively, and l_1, l_2 are 2D position parameters. The transformation of each parameter is independently represented by a scalar Gaussian distribution around their previous statess_{t-1}. Then the motion model is obtained by a Gaussian distribution as follows:

$$p(s_t \mid s_{t-1}) = \mathbb{N}(s_t; s_{t-1}, N), \tag{11}$$

where $\mathbb{N}(\cdot)$ is the Gaussian distribution and N is the covariance matrix. Therefore, the observation model $p(y_t|s_t)$ can be defined by

$$p(y_t \mid s_t) \propto \sum_{s}^{L} \sum_{i}^{K} F_i^s, \tag{12}$$

where the right side of the equation denotes the similarity between the candidate and the target based on the pooled feature F_i^s defined in Eq. (7).

4.2 Two-step object tracking

As we know, the ground truth plays a key in determining whether a new tracking result is effective during the tracking process. Since there is no ground truth available in practical applications, noise inevitably occurs when updating the observation model with multi-scale dictionary in this work. It could degrade the linear representation of the target patch by the multi-scale dictionary, thereby causing tracking drift gradually. To solve the problem, a novel two-step algorithm is proposed involving the use of dynamical appearance model and static appearance model based on multi-scale patch. The dynamical model is represented by sparse coefficients of different scales and is updated online by the multi-scale dictionary, while the static one is represented by sparse coefficients under different scales with dictionary from the first frames. Thus, a dynamical likelihood function and a static likelihood function can be respectively constructed by the two models using Eqs. (7) and (12). A similar strategy has also been successfully applied in the reduction of tracking drift [12].

In practical cases, the ground truth is the region of labeled target image in the first frames. Initially, we construct a static appearance model involving static dictionary D_0^s based on different scale patches and it is used to compute the static likelihood function in the second step. At time t, target candidate positions are estimated using dynamical likelihood function involving the online updated dictionary D_t^s within particle filtering framework. Subsequently, with the static appearance model involving static dictionary D_0^s, each image patch p_i^s of the estimated tracking result, based on different scale s, will have a corresponding vector of reconstruction coefficients α_i^s, which is computed by:

$$\hat{a}_i^s = \arg\min \left\|\alpha_i^s\right\|_1 \text{ subject to } \left\|p_i^s - D_0^s\alpha_i^s\right\|_2 < \varepsilon. \quad (13)$$

Finally, sparse coefficients α_i^s are used to compute the static likelihood function by Eqs. (7) and (12) and the final target position is determined by the static likelihood function. In

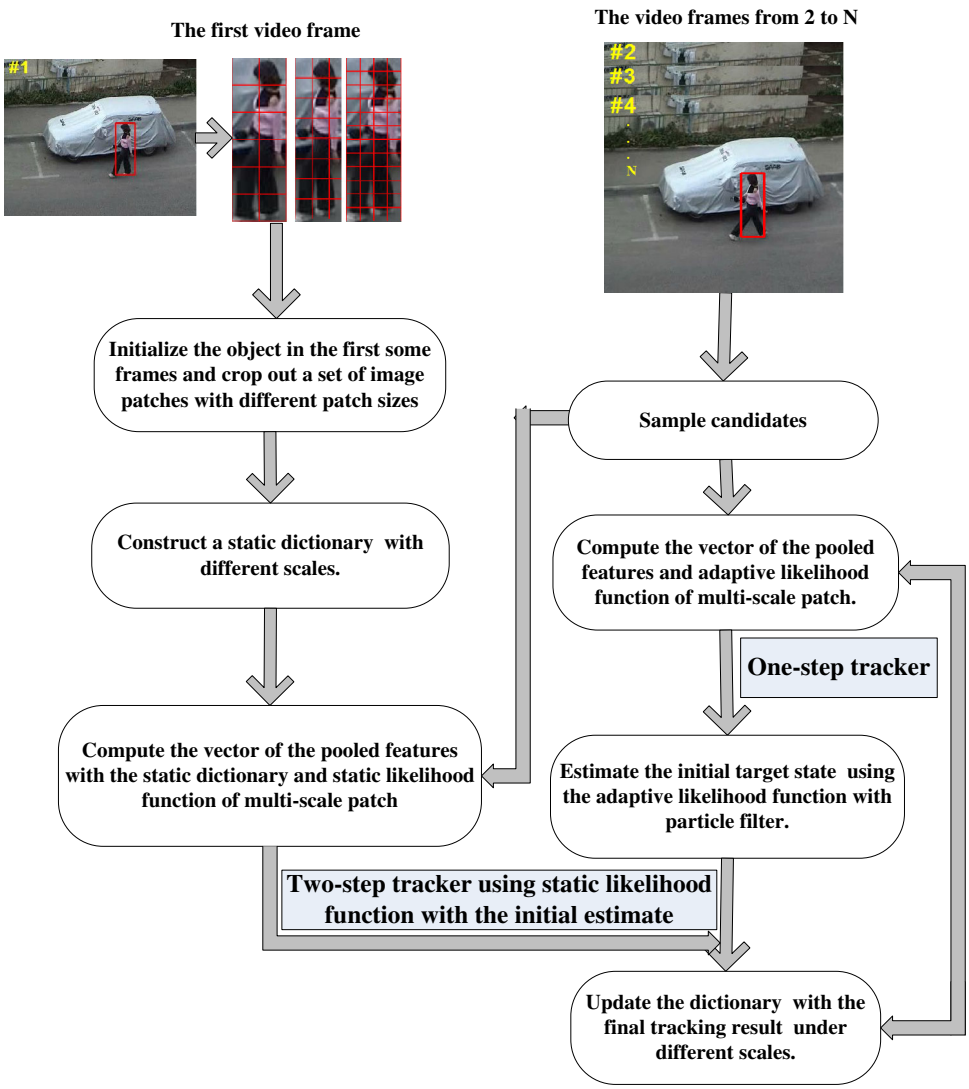


Fig. 3 Flowchart of the proposed two-step tracking algorithm

a conclusion, in the proposed two-step tracking strategy, the first stage can capture very large appearance changes and create a number of candidate positions by dynamical likelihood function with particle filter; the second stage selects the best candidate position and ensures the final tracking result as similar to the ground truth obtained in the first frames by static likelihood function. The flowchart of the proposed two-step tracking algorithm is shown as Fig. 3.

4.3 Update scheme

The appearance of an object may be changed drastically while undergoing significant pose change, illumination variation or partial occlusion in the tracking process. To reflect the changes, online updated dictionary D^s is applied in the dynamical appearance model every five frames. When the tracking result at time t is obtained, the corresponding target observation $P^s = \{p_i^s | i = 1 : K, s = 1, 2, ..., L\}$ is used to update D^s with the consideration of reconstruction error of the local patch. The dictionary D^s under scale s is updated by $D^s = \{p_j^s | j = (n-l) \times K : (n-l+1) \times K, s = 1, 2, ..., L, r = 2 \times s + 2\}$ when $\beta^s > 0.5$, where l is a random number with $0 < l < n$.

The proposed tracking algorithm is summarized in Algorithm 1.

The proposed tracking algorithm is summarized in algorithm 1.

1: **Input**: The initial state of the target $s_1 = (l_1, l_2, \mu_1, \mu_2, \mu_3, \mu_4)$, video frames $F_1, ...F_t$, a static dictionary D_0^s based on different scale patches from the first frames.

2: **Output:** The current target state s_t at time *t*.

Initialization:

3: Initialize the object in the first frame and crop out a set of image patches with different patch sizes.

4: Construct a static dictionary D_0^s with different scales.

Online tracking:

5: for $t = 2, ..., T$ do

6: for $s = 1, 2, ..., L$ do

7: Sample candidates and calculate the corresponding sparse coefficients under different patch sizes with the dictionary D_{t-1}^s by Eq. (6).

8: Compute the vector of the pooled features F^s and dynamical likelihood function of multi-scale patches using Eqs. (7) and (12), respectively.

9: Compute the vector of the pooled features F_0^s with the static dictionary D_0^s and static likelihood function of multi-scale patches using Eqs. (6), (7) and (12).

10: End for.

11: First-step tracker: Estimate the initial target state $s_t^{'}$ using the dynamical likelihood function with particle filter.

12: Second-step tracker: Determine the final tracking result s_t using the static likelihood function with the initial estimate $s_t^{'}$.

13: Update the dictionary D_t^s with the final tracking result s_t under different scales.

14: End for.

Table 1 The tracking sequences used in our experiments

Video name	Number of frames	Main challenges
Woman	550	Partial occlusion, pose change
Shop2cor	350	Heavy occlusion, pose change
Faceocc2	814	Partial occlusion, in-plane pose change
David	462	Illumination variation, in-plane/out-of-plane pose change, partial occlusion
Dylv	1,344	In-plane/out-of-plane pose change, fast motion, illumination change
Dirl	502	Partial occlusion, fast motion, in-plane change, moving camera
Lemming	1,336	Heavy occlusion, very fast motion, in-plane/out-of-plane pose change
Box	1,161	Heavy occlusion, very fast motion, pose change

5 Experiments

To investigate the performance of our proposed method, experiments on eight publicly available video sequences are conducted involving the challenges of partial or significant occlusion, camera moving, pose and illumination changes and so on. The details of the selected video sequences are listed in Table 1. Also, five state-of-the-art trackers are tested on the same sequences, including incremental visual tracking (IVT tracker) [15], L1 tracking (L1 tracker) [5], adaptive structural local sparse appearance model (ASLSAM tracker) [10], partial least squares analysis (PLS tracker) [12] and partial real-time compressive tracking (CT tracker) [35]. For fair comparison, all of them were experimented on the same dynamic model and the same particles (300 particles per frame in this work) and used the same initialized target locations in theses video sequences. The tracking videos, MATLAB codes and data sets can be respectively found from URLs [36–38]. Seven scales with patch sizes 4×4, 6×6, 8×8, 10×10, 12×12, 14×14 and 16×16 are used in our work and the size of the sampled image patch is set to 32×32. Moreover, all of these experiments are conducted using a MATLAB implementation on a 2.5 GHz machine with 4 GB RAM.

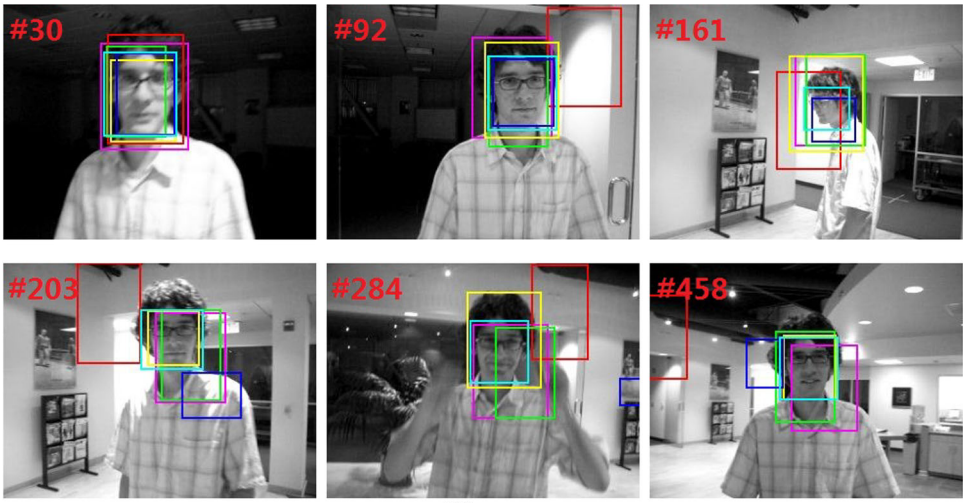
For this method, the computational complexity is dominated by sparse approximation for representing the target. The Orthogonal Matching Pursuit (OMP) [39] is applied for seeking the sparsest linear combination efficiently. The computational cost of OMP is much less than that of the $\ell1$-norm minimization with standard convex programming for sparse representation problems. By using the OMP to search for sparse coefficients, neglecting the cost of least-square steps, each image patch can be found in $O(nKd^s)$ operations under scale s, which thereby costs $O(nK^2d^s)$ operations for the whole sparse coding stage. Statistically, our algorithm runs at around 5 s per frame and the IVT, L1, ASLSAM, PLS and CT trackers spend about 0.5, 10, 1.5, 1.2 and 0.8 s per frame, respectively. The most time consuming part of our tracker is the computation of sparse coefficients using the dictionary under different scales. Therefore, it is possible that the efficiency of our method could be further increased by reducing the data dimension via feature extraction such as k-means cluster and principal component analysis (PCA) [15], replacing the ℓ 1-minimization in this work by block orthogonal matching pursuit (BOMP) [8]. In addition, the number of different scales is a trade-off between computational efficiency and effectiveness of modeling target appearance changes. To further reduce the computational load of sparse coefficients under different scales, a limited number of scales (e.g., 4×4, 8×8, 12×12, or 16×16) could be adopted for object representation, in which scales are insensitive to rotation, scale variation and complex background in reality. For qualitative analysis, some representative frames are selected to show the evaluation comparison of our tracker and the others.

5.1 Qualitative analysis

To make methods more robust and efficient, the initial values of l_1, l_2will be set to be larger when the target location change is very obvious between two consecutive frames, such as cases in the sequence "lemming" and vice verse. The initial of μ_1 will be set to a larger value when the tracked targets encounter greater rotation between two consecutive frames, such as cases in the sequence "david" and "girl" and vice verse. The initial values of μ_2, μ_3 will be smaller when the tracked targets undergo smaller-scale and aspect ratio change during the tracking process and vice verse. Similarly, the initial value of μ_4 is smaller when the tracked targets undergo smaller skew direction change during the tracking process and vice verse. Therefore, for the sequences "Woman", "Shop2cor", "faceocc2", "david", "sylv" "girl", "lemming" and "box", the variances of affine parameters s_1 are set to (4, 4, 0.01, 0, 0, 0), (4, 4, 0.01, 0, 0.001, 0), (4, 4, 0.02, 0, 0.001, 0), (3, 3, 0.02, 0, 0.01, 0), (10, 10, 0.005, 0, 0.001, 0), (10, 10, 0.002, 0, 0.001, 0), (10, 10, 0.02,

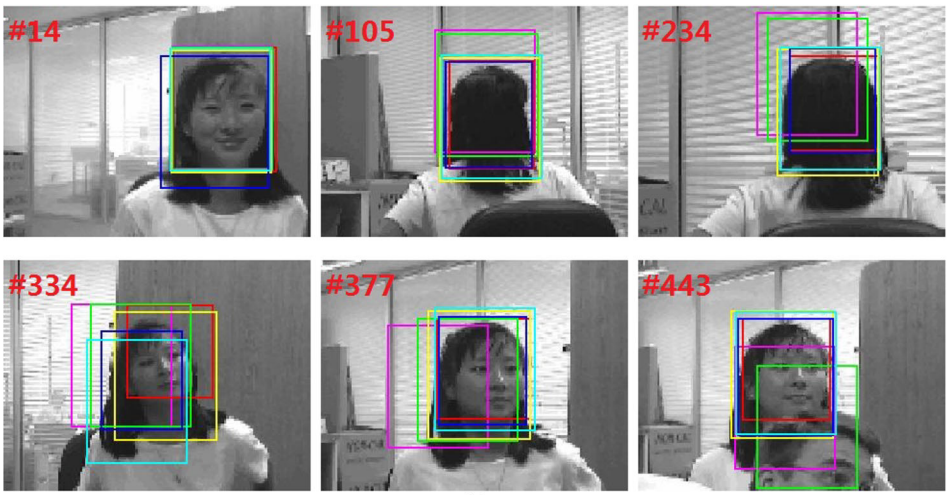
Fig. 4 Screenshots of tracking comparison of our tracker (*yellow box*) with L1 tracker (*red box*), IVT tracker (*mulberry box*), CT tracker (*green box*), PLS tracker (*blue box*) and ASLSAM tracker (*cyan box*), highlighting instances of partial occlusion, illumination variation, heavy occlusion, fast motion, in-plane/out-of-plane pose change and moving camera

Fig. 4 continued



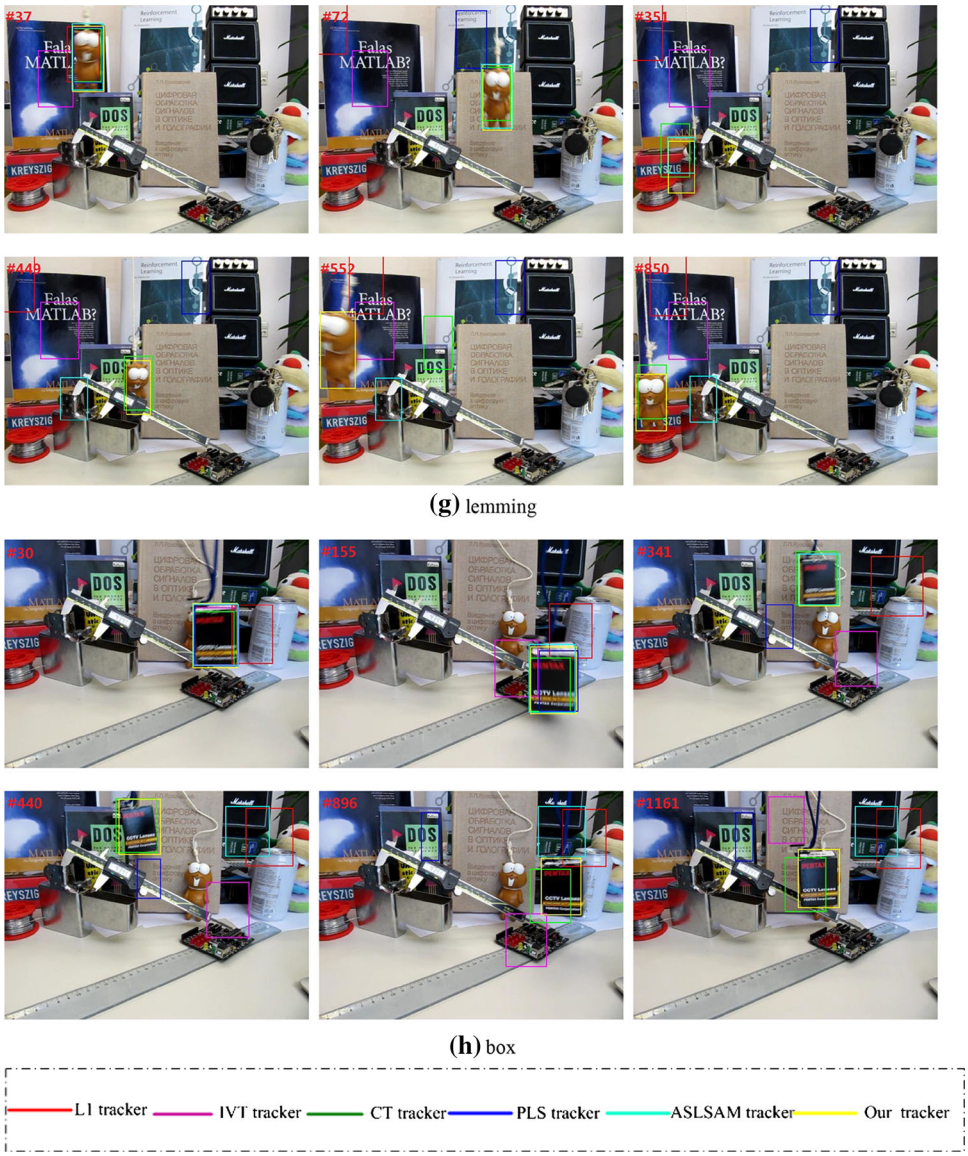
(d) david

(e) sylv



(f) girl

Fig. 4 continued



0, 0.02, 0), (10, 10, 0.02, 0, 0.02, 0), respectively, in this experiment.

As seen in Fig. 4a, our tracker and the ASLSAM tracker show competitive performance for the whole sequence frames compared with L1, IVT, PLS and CT trackers. However, the target starts to drift in frames 81 and 132 by the L1 and IVT tracker. Except for our tracker and ASLSAM tracker, other four trackers fail in capturing target in frames 199, 313 and 493. As illustrated in Fig. 4b, our tracker performs the same with the ASLSAM and PLS trackers and can track the man successfully for the whole sequence, while CT tracker appears to target drifting in frame 79 and the L1 and IVT trackers totally lose the target from the beginning of the frame 100. As illustrated in Fig. 4c, all of them can track the target successfully for the whole sequence, while PLS tracker appears to target drifting in frames 490, 654 and 802. From Fig. 4d, L1 tracker loses the target from the early frames of the sequence due to a sudden illumination variation. Tackers of PLS, IVT and CT cannot track the target and drift from the target area in frames 203, 284 and 458. In Fig. 4e, L1 tracker fails to track the target after frame 440 and, PLS and ASLSAM trackers also miss the target in frame 634 and 755. Surprisingly, both our tracker and the IVT tracker yield satisfactory performance. As shown in Fig. 4f, our tracker achieves the best performance during the whole sequence. Moreover, L1, PLS and ASLSAM trackers perform good, while IVT and CT trackers drift away from the target area in frames 234, 334, 377 and 443. From Fig. 4g, L1, IVT and PLS drift away from the target area very quickly because of too much fast motion of the target, while the CT tracker and ASLSAM tracker fail in target tracking in

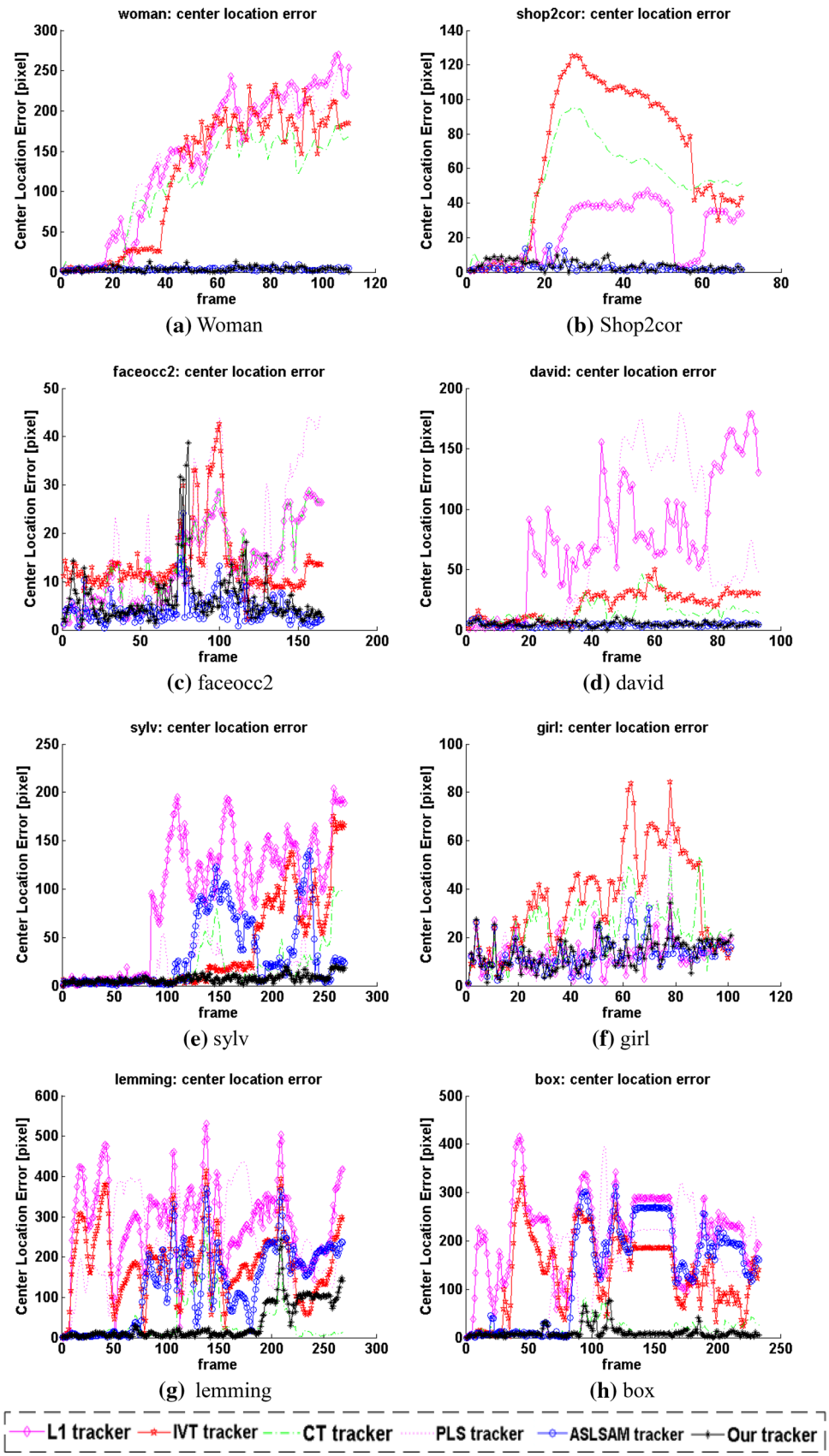
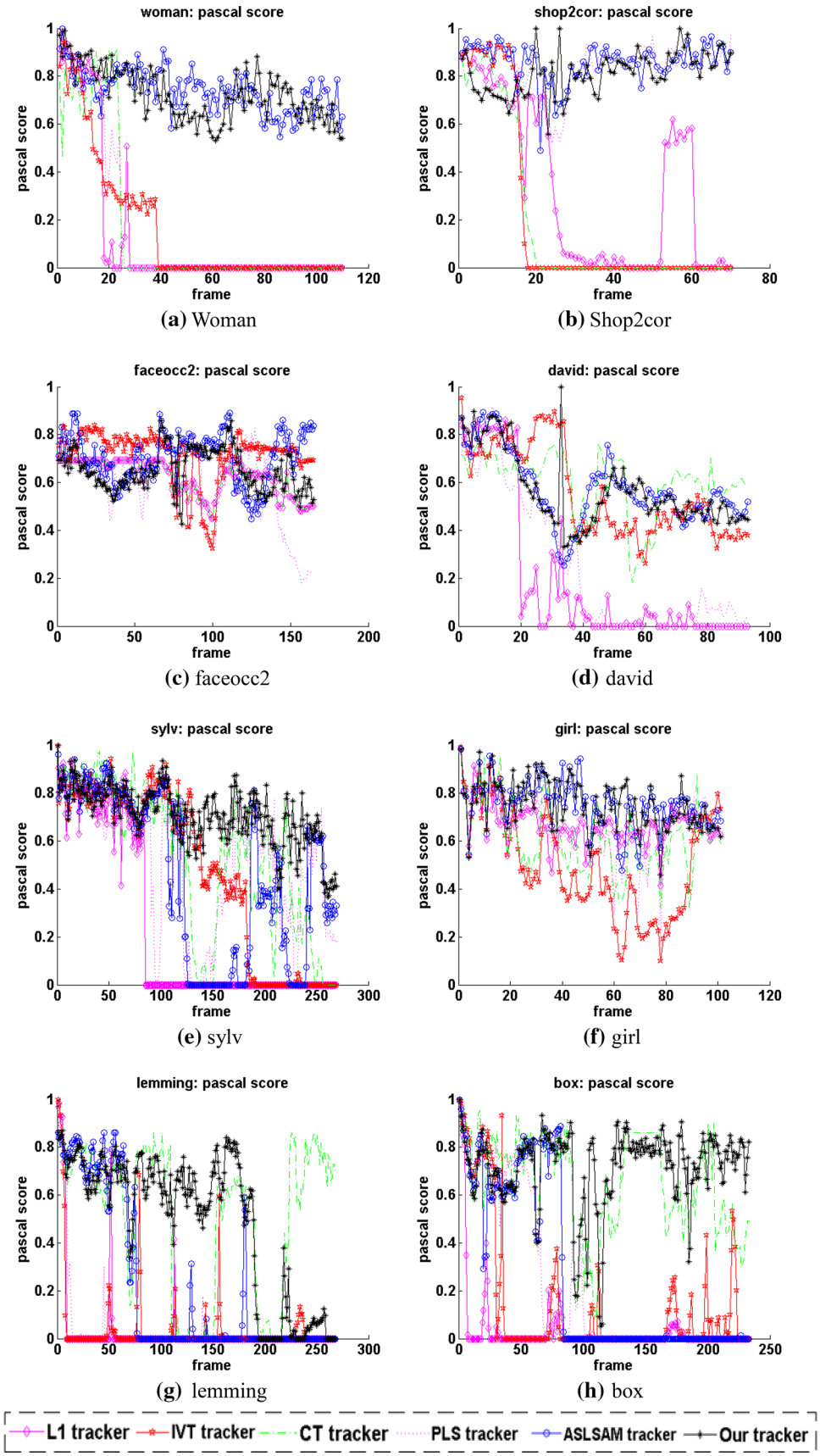


Fig. 5 Center location error plots for our tracker, L1 tracker, IVT tracker, CT tracker, PLS tracker and ASLSAM tracker



(a) Woman **(b)** Shop2cor **(c)** faceocc2 **(d)** david **(e)** sylv **(f)** girl **(g)** lemming **(h)** box

Fig. 6 Pascal score plots for our tracker and the other five trackers

Table 2 Center location errors (pixel) of our tracker with L1, IVT, CT, PLS and ASLSAM trackers

Video name	L1 tracker	IVT tracker	CT tracker	PLS tracker	ASLSAM tracker	Our tracker
Woman	142.5	119.6	109.1	137.9	*3.3*	**2.0**
Shop2cor	23.5	65.8	50.7	2.9	**2.6**	*2.7*
Girl	**12.9**	35.5	23.9	15.3	*13.2*	13.3
Sylv	91.9	36.4	19.6	*16.3*	31.6	**8.8**
David	75.4	20.5	15.3	63.9	*6.6*	**5.0**
Faceocc2	14.6	14.9	13.8	15.4	*6.1*	**5.1**
Lemming	280.9	183.9	*40.4*	241.4	95.3	**30**
Box	209.1	152.5	*18.7*	137.3	134.2	**8.3**
Overall center location errors	850.8	629.1	*291.5*	630.4	293.1	**75.2**

Bold number indicates the best performance; italicized number indicates the second best tracker for each sequence

Table 3 Pascal scores of our tracker and the other compared trackers demonstrate the success rate of the successfully tracked frames for each sequence

Video name	L1 tracker (%)	IVT tracker (%)	CT tracker (%)	PLS tracker (%)	ASLSAM tracker (%)	Our tracker (%)
Woman	16.7	13.1	21.5	18	*79.5*	**87.8**
Shop2cor	42.9	21.4	22.9	**100**	*98.6*	*98.6*
Girl	*98*	44.6	77.2	89.1	*98*	**99**
Sylv	30.9	52	*61*	59.5	52	**84.7**
David	20.4	46.2	*82.8*	25.8	72	**97.8**
Faceocc2	*92*	91.4	*92*	74.2	**96.9**	*92*
Lemming	5.2	6	**66.8**	4.9	26.5	*66.5*
Box	3	13.3	*73.4*	26.6	32.2	**93.4**
Average Pascal score	38.7	36.1	59.5	49.8	*69.5*	**89.9**

For each sequence, bold number indicates the best performance; italicized number indicates the second best tracker

frame 552. Our tracker can track the target well in the whole sequence. In Fig. 4h, we found that the L1 tracker and IVT tracker fail to track the target after frame 155 due to its very fast motion. The ASLSAM tracker and PLS tracker also fail in frames 341, 440, 896 and 1161. Comparatively, our tracker and the CT tracker perform well in the whole sequence.

5.2 Quantitative analysis

Two criteria are used to quantitatively evaluate the performance of our proposed tracker. Center location error measures the Euclidean distance between the central position of the tracking result and that of the manually labeled ground truth. In our experiments, the ground truth centers of the objects in woman, Shop2cor, faceocc2, david, sylv, girl, lemming and box video clips for every five frames are provided by [10] and [38]. Similarly to [40], the success rate is the second criterion that indicates the number of successful tracked frames and is defined as:

$$\text{Pascal score} = \frac{B_R \cap B_T}{B_R \cup B_T}, \tag{14}$$

where B_R and B_T are the tracked bounding box and the ground truth bounding box, respectively. It is defined that a tracking result in one frame is considered as a success when the Pascal score is above 0.5. Figures 5 and 6 illustrate the comparison of our tracker with L1, IVT, ASLSAM, CT and PLS trackers with respect to the two criteria, and the corresponding center location errors and Pascal scores are listed in Tables 2 and 3, respectively. Experimental results showed that our proposed tracking algorithm outperforms the others on the "Woman", "sylv", "david", "faceocc2", "lemming" and "box" sequences. As shown in Table 2, our method has the lowest overall center location errors, implying that it is more robust than the other five trackers. From Table 3, we also observe that our tracker achieves the highest success rate compared with the other five trackers except for the cases of "Shop2cor", "faceocc2" and "lemming" sequences. Moreover, our tracker achieves an average score of 89.9 %, which indicates that it Wednesday, August 13, 2014 at 9:52 amis of

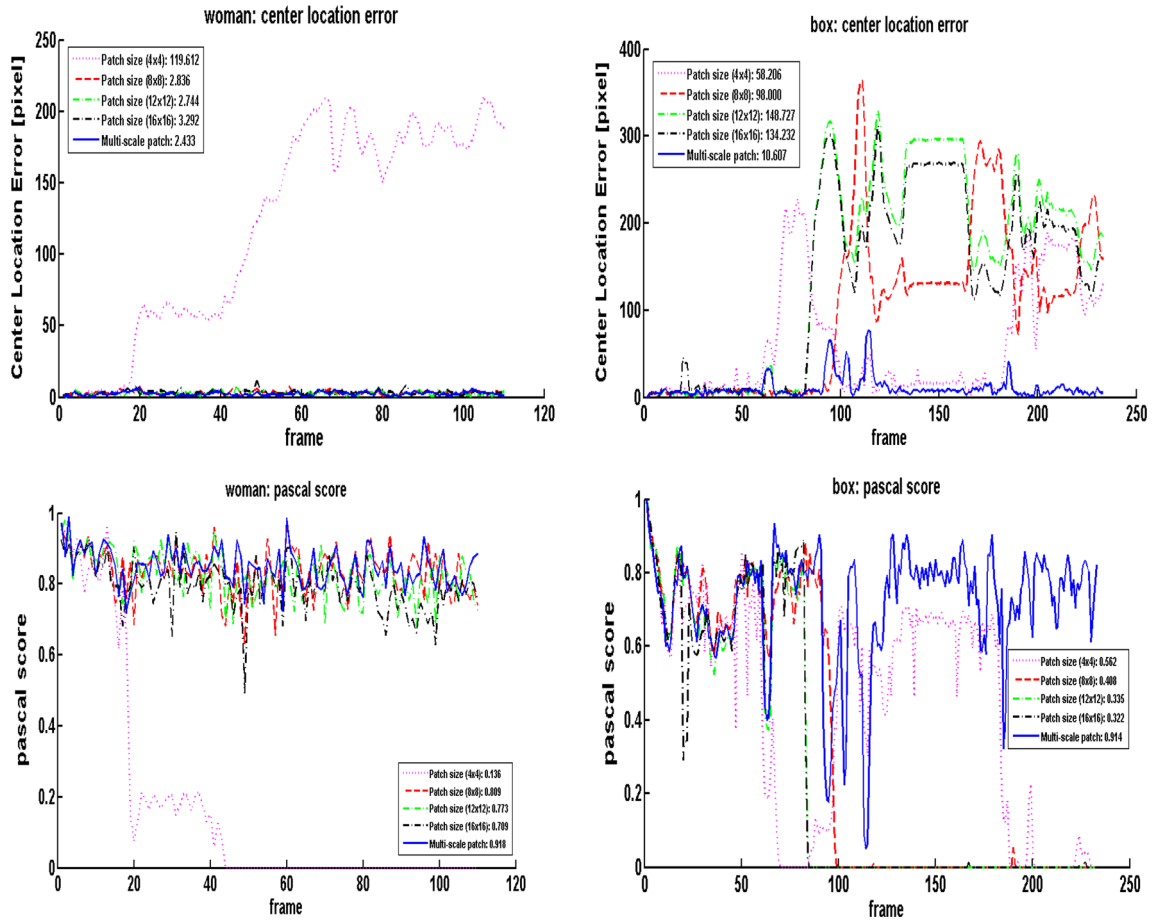


Fig. 7 Center location error and Pascal score comparison on the "Woman" and "box" sequences for different scales

the highest success rate than the other trackers in the eight experiments.

5.3 Case studies of tracking under different scales

Figure 7 demonstrates the comparison of center location error and success rate on "Woman" and "box" video sequences under different patch sizes. Four scales with the patch size $4 \times 4, 8 \times 8, 12 \times 12$ and 16×16 are tested here. From Fig. 7, different patch sizes yield different results on the "Woman" and "box" sequences and the patch size is sensitive for the tracking scenarios. However, our multi-scale patch-based tracker performs more robustly than those with single-scale patch because of the fact that patches on different scales contain complementary information for tracking.

5.4 Two-step tracking analysis

To further demonstrate the power of our two-step strategy for target tracking, a one-step tracker is constructed with the online updated dictionary and the evaluation on the eight video sequences is implemented. Figure 8 shows the success rate comparison of the one-step tracker and the two-step tracker on the eight video sequences. From Fig. 8, we found that our two-step tracker performs more robustly in success rate than the one-step tracker, which lacks static appearance model. In summary, for the two-step tracking strategy, the first stage can avoid the local minimum problem effectively and capture very large appearance change between two consecutive frames; the second stage can ensure the final tracking result as similar to the ground truth since it integrates the ground truth information from the first frame.

6 Conclusion

To track visual object undergoing various appearance changes, in this paper we propose a novel, robust and dynamical approach with the design of appearance model, i.e., multi-scale patch-based sparse representation. Since patch size greatly influences the final tracking result, different

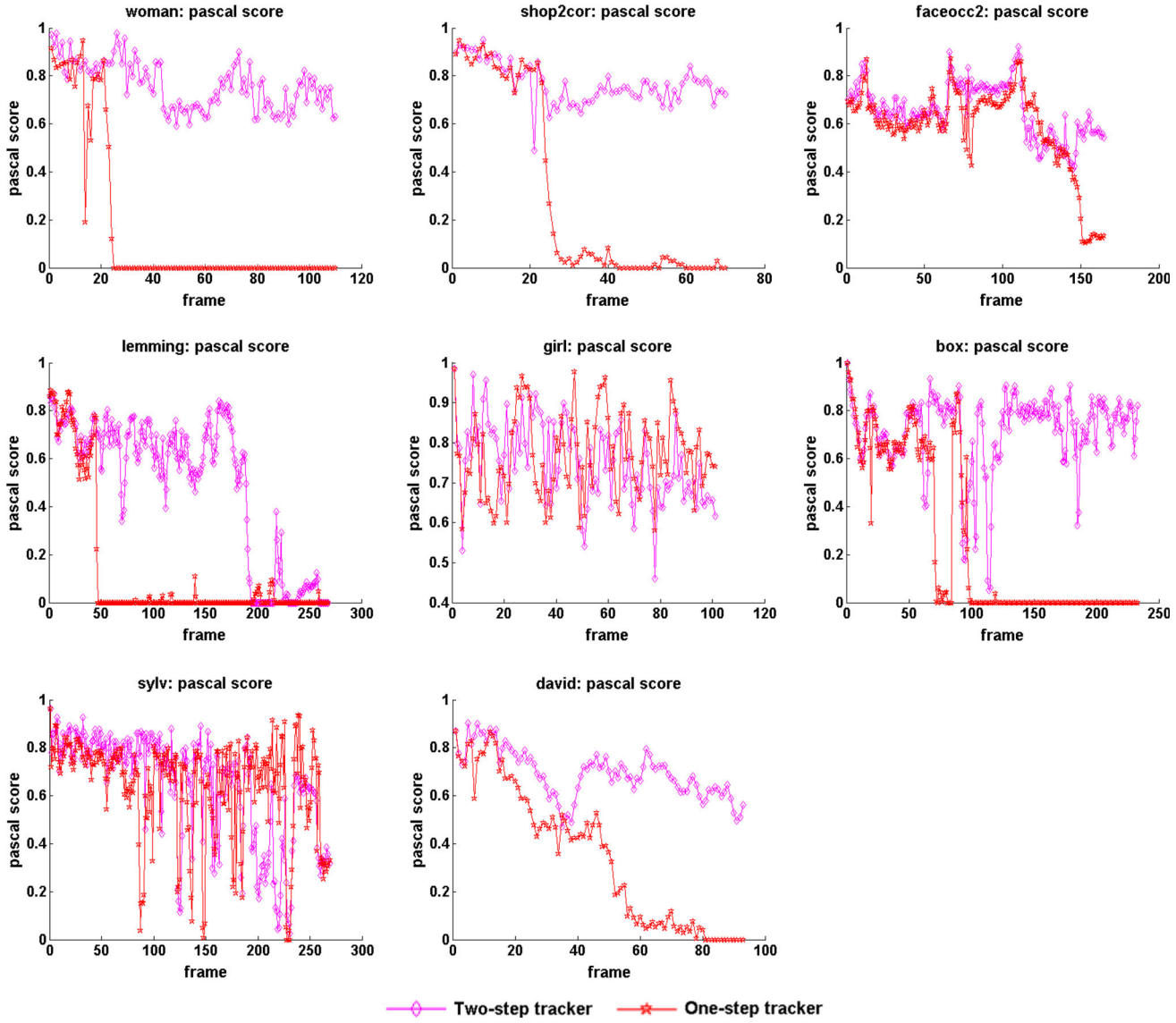


Fig. 8 Pascal score comparison of the one-step tracker (without the static appearance model) with the two-step tracker (with the static appearance model) on eight video sequences

from traditional local sparse representation model with fixed patch size, this work adopts multiple patch sizes and then uses sparse coefficients of multi-scale patches to define a dynamical likelihood function by the alignment-pooling method. Finally, the dynamical likelihood function is embedded into a Bayesian inference framework for the estimation of the initial object state in consecutive frames. To reduce tracking drift, with the initial object state, a static likelihood function under different scales is integrated into Bayesian inference framework again for obtaining the final tracking result. Experiments on some challenging video sequences show that our proposed tracker achieves state-of-the-art performance in both qualitative and quantitative respects.

Acknowledgments This work was supported by the NSFC-Guangdong Joint Foundation Key Project under Grant (No. U1135003), the National Nature Science Foundation of China (Nos. 61070227, 61300058 and 41302261).

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Survey **38**(4), 1–45 (2006)
2. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: Sparse representation for computer vision and pattern recognition. Proc. IEEE. **98**(6), 1031–1044 (2010)
3. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Comm. Pure Appl. Math. **59**(8), 1207–1223 (2006)

4. Donoho, D.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
5. Mei, X., Ling, H.: Robust visual tracking using l_1 minimization, ICCV, pp. 1436–1443 (2009)
6. Li, H.X., Shen, C.: Robust real-time visual tracking using compressive sensing. CVPR, pp. 1305–1312 (2011)
7. Han, Z., Jiao, J., Zhang, B., Ye, Q., Liu, J.: Visual object tracking via sample-based adaptive sparse representation (AdaSR). Pattern Recogn. **44**(9), 2170–2183 (2011)
8. Bai, T., Li, Y.F.: Robust visual tracking with structured sparse representation appearance model. Pattern Recogn. **45**(6), 2390–2404 (2012)
9. Liu, B., Huang, J. et al.: Robust tracking using local sparse appearance model and K-selection. CVPR, pp. 1313–1320 (2011)
10. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. CVPR, pp. 1822–1829 (2012)
11. Matthews, I., Ishikawa, T., Baker, S.: The template update problem, PAMI. 810–815 (2004)
12. Wang, Q., Chen, F., Xu, W., Yang, M.-H.: Object tracking via partial least squares analysis. IEEE Trans. Image Process. **21**(10), 4454–4465 (2012)
13. Pérez, P., Hue, C. Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Proceedings of European conference on computer vision, pp. 661–675 (2002)
14. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 564–575 (2003)
15. Ross, D., Lim, J., Lin, R.S., Yang, M.-H.: Incremental learning for robust visual tracking. IJCV **77**(1), 125–141 (2008)
16. Kwon, J., Lee, K. M.: Visual tracking decomposition. CVPR, pp. 1269–1276 (2010)
17. Kwon, J., Lee, K.M.: Tracking by sampling trackers. ICCV, pp. 1195–1202 (2011)
18. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1631–1643 (2005)
19. Ramirez-Paredes, J.-P., Sanchez-Yanez, R.E., Ayala-Ramirez, V.: A fuzzy inference approach to template-based visual tracking. Mach. Vis. Appl. **23**(3), 427–439 (2012)
20. Wang, D., Lu, H.C., Yang, M.-H.: Online Object Tracking with sparse prototypes. IEEE Trans. Image Process. **22**(1), 314–325 (2013)
21. Zhong, W., Lu, H.C., Yang, M.-H.: Robust object tracking via sparsity-based collaborative model. In CVPR, pp. 1838–1845 (2012)
22. Wu, Y., Cheng, J., Wang, J.Q., Lu, H.Q., Wang, J., Ling, H.B., Blasch, E., Bai, L.: Real-time probabilistic covariance tracking with efficient model update. IEEE Trans. Image Process. **21**(5), 2824–2837 (2012)
23. Avidan, S.: Ensemble tracking. PAMI **29**(2), 261–271 (2007)
24. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS. 1417–1426 (2005)
25. Avidan, S.: Support vector tracking. In: CVPR. pp. 184-191 (2001)
26. Grabner, H., Bischof, H.: On-line boosting and vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 260–267 (2006)
27. Parag, T., Porikli, F., Elgammal, A.: Boosting adaptive linear weak classifiers for online learning and tracking. CVPR. pp. 1–8 (2008)
28. Carvalho, P., Oliveira, T., Ciobanu, L., Gaspar, F., et al.: Analysis of object description methods in a video object tracking environment. Mach. Vis. Appl. **24**(6), 1149–1165 (2011)
29. Babenko, B., Yang, M.-H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
30. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2008)
31. Chen, F., Wang, Q., Wang, S., Zhang, W., Xu, W.: Object tracking via appearance modeling and sparse representation. Image Vis. Comput. **21**(11), 787–796 (2011)
32. Wang, Q., Chen, F., Xu, W., Yang, M.-H.: Online discriminative object tracking with local sparse representation', WACV '12 Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision, pp. 425–432 (2012)
33. Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
34. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)
35. Zhang K.H., Zhang L., Yang M.H.: Real-time compressive tracking. ECCV, pp. 864–877 (2012)
36. http://www.dabi.temple.edu/~hbling/code_data.htm
37. http://www.cs.toronto.edu/~dross/ivt/
38. http://faculty.ucmerced.edu/mhyang/pubs.html
39. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**(12), 4655–4666 (2007)
40. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. IJCV **88**(2), 303–338 (2010)

Chengjun Xie received his master's degree in software engineering from the Hefei University of Technology, China, in 2008. Since 2011, he has been a Ph.D. candidate of Hefei University of Technology, Hefei, China. His research interests include image processing, intelligent surveillance, etc.

Jieqing Tan received his B.S. degree in computational mathematics at Xi'an Jiaotong University, Xi'an, China in 1984, and M.S and Ph.D. degrees from Hefei University of Technology and Jilin University, China in 1987 and 1990, respectively, all in computational mathematics. His current research interests include non-linear numerical approximation, computer graphics and image processing.

Peng Chen is currently a professor at the Institute of Health Sciences, Anhui University, Hefei, China. He received his bachelor's and master's degrees in control science and engineering from the Electronic Engineering Institute and Kunming University of Science and Technology, respectively, and Ph.D degree in pattern recognition and intelligent system from the University of Science and Technology of China. From 2006 to 2013, he was senior research associate at the City University of Hong Kong, postdoc fellow at Howard University in USA, research fellow at Nanyang Technological University in Singapore and postdoc fellow at King Abdullah University of Science and Technology in Saudi Arabia, respectively. Dr. Chen's research interests include machine learning and data mining with applications in bioinformatics, etc.

Jie Zhang received his master's degree in software engineering from the Hefei University of Technology, China, in 2009. Since 2010, he has been a Ph.D. candidate of the University of Science and Technology of China, Hefei, China. His research interests include robot vision and machine learning.

Lei He received his master's degree in software engineering from the School of Computer and Information, Hefei University of Technology, China, in 2007. Since 2011, he has been a Ph.D. candidate of Hefei University of Technology, Hefei, China. His research interests include image processing, pattern recognition, etc.