www.ietdl.org

Published in IET Computer Vision
Received on 11th October 2012
Revised on 6th April 2013
Accepted on 13th May 2013
doi: 10.1049/iet-cvi.2012.0228

ISSN 1751-9632

Multiple instance learning tracking method with local sparse representation

Chengjun Xie[1,2], Jieqing Tan[1], Peng Chen[2,3], Jie Zhang[2], Lei He[1]

[1]*School of Computer and Information, Hefei University of Technology, Hefei 230009, People's Republic of China*
[2]*Laboratory of Intelligent Decision, Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, People's Republic of China*
[3]*Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*
E-mail: pchen.ustc10@gmail.com

Abstract: When objects undergo large pose change, illumination variation or partial occlusion, most existed visual tracking algorithms tend to drift away from targets and even fail in tracking them. To address this issue, in this study, the authors propose an online algorithm by combining multiple instance learning (MIL) and local sparse representation for tracking an object in a video system. The key idea in our method is to model the appearance of an object by local sparse codes that can be formed as training data for the MIL framework. First, local image patches of a target object are represented as sparse codes with an overcomplete dictionary, where the adaptive representation can be helpful in overcoming partial occlusion in object tracking. Then MIL learns the sparse codes by a classifier to discriminate the target from the background. Finally, results from the trained classifier are input into a particle filter framework to sequentially estimate the target state over time in visual tracking. In addition, to decrease the visual drift because of the accumulative errors when updating the dictionary and classifier, a two-step object tracking method combining a static MIL classifier with a dynamical MIL classifier is proposed. Experiments on some publicly available benchmarks of video sequences show that our proposed tracker is more robust and effective than others.

1 Introduction

Object tracking is a well-studied issue in computer vision and plays a crucial role in many practical applications, such as video surveillance, human motion understanding and interactive video processing and so on. Although existing trackers have made some success under various scenarios, objects tracking is still challenging because the appearance of an object can be changed drastically while it undergoes significant pose change, illumination variation and/or partial occlusion. Such a thorough review can be found in [1], which presented a typical tracking system that can be decomposed into 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 for 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 integrating online multiple instance learning (MIL) with local sparse representation.

In [2], the tracking problem was formulated to find a sparse approximation using template subspace, and experiments were also found to be efficient and adaptive to address the aforementioned challenges, especially in the case of partial occlusion. However, besides the high computational cost of the tracking process, another drawback is the limitation of the appearance method to model holistic object appearance within a generative framework. The results have shown that training an adaptive model to separate object from the background by a discriminative classifier can often obtain good tracking results. However, a major challenge of the discriminative method is how to choose positive and negative samples when updating the adaptive appearance model. Moreover, most discriminative trackers took the current object location as one positive sample, and sampled its neighbourhoods for negatives. If the current object location is imprecise, however, this could degrade the appearance model and cause drift. On the other hand, it is very difficult to discriminate the object when updating classifier by the use of multiple positive samples.

To overcome those challenges, we turn to adopt a discriminative learning paradigm called MIL. The pioneer work of MIL was reported in [3], in which labels of training data were naturally represented by a bag of instances instead of individual one. In the framework of MIL, a bag is labelled as positive if it contains at least one positive instance, otherwise the bag is negative. Actually, a positive bag may contain a few possible bounding boxes around the current object location, whereas the MIL can effectively eliminate the ambiguity (the unwanted instances) and figure out which instance in each positive bag is the most important one (the most correct instance). Subsequently, Babenko *et al.* [4, 5] developed an

online MIL algorithm for visual tracking. Although their results showed that the MIL can help reduce drifts for object tracking, the instances in bags cannot be selected effectively because of the noise caused by frequently updating the MIL appearance model with new input targets to be tracked. To decrease the visual drift, Matthews *et al.* [6] proposed a two-step approach to model appearance for robust tracking target, whose first step was to estimate tracking result by applying online template from the most recent frame and then the final target location was determined by using the template from the first frame. However, it also suffered from drift problem when an object undergoes partial occlusion. Wang *et al.* [7] proposed a two-stage algorithm to exploit both the ground truth information of the first frame and observations obtained online. However, it modelled target as a single instance, trained a liner classifier by sparse codes and thus faced the same challenge similar to that mentioned above, in that it was difficult to precisely choose positive and negative samples when updating the liner classifier.

Inspired by the works mentioned above, we propose an efficient tracking algorithm containing online MIL based on local sparse representation. On the one hand, the unwanted instances can be removed by the MIL trainer, which can strengthen the MIL classifier with true positive instances. On the other hand, local sparse representation has been shown to be more robust than other representations when objects undergo pose change, illumination and/or partial occlusion. Therefore different from traditional discriminative tracking methods that use multiple image features, this paper proposes a more robust appearance model by making full use of the advantages of the MIL and local sparse representation. First, an overcomplete dictionary is learned directly from raw image patches. Then, objects are represented with local sparse codes by minimising the reconstruction error and maximising the sparsity of each image patch at the same time. Finally, the MIL classifier is trained by the local sparse codes of the positive and negative image patches shown in Fig. 1 and the tracking task is formulated as a classification problem.

Here the proposed two-step tracking strategy adopts a dynamic and a static MIL classifier based on local sparse representation, which can effectively select the wanted instances and eliminate the unwanted instances in bags to train MIL classifier for robust tracking. In the first step, sparse codes of each candidate are computed using the online updated dictionary, and then a dynamical classifier is applied to discriminate the target and estimates its candidate locations. In the second step, a static MIL classifier is trained by using the sparse codes of image patches consisting of positive and negative ones from the first video frame, and then it is used to select the best accurate target position resulted from the first step. The flowchart of the proposed method is shown in Fig. 2.

In summary, the first contribution of this paper is the proposed adaptive appearance model combining online MIL and local sparse representation, which makes full use of the advantages of MIL and local sparse representation. On the one hand, local sparse representation has more robust appearance model, especially in the case of partial occlusion. The objects can be effectively represented by sparse codes of the corresponding image patches. On the other hand, the MIL classifier can handle ambiguous binary data obtained online. Therefore different from [5] whose training data for the MIL classifier are composed of sparse codes instead of a pool of features directly from raw image

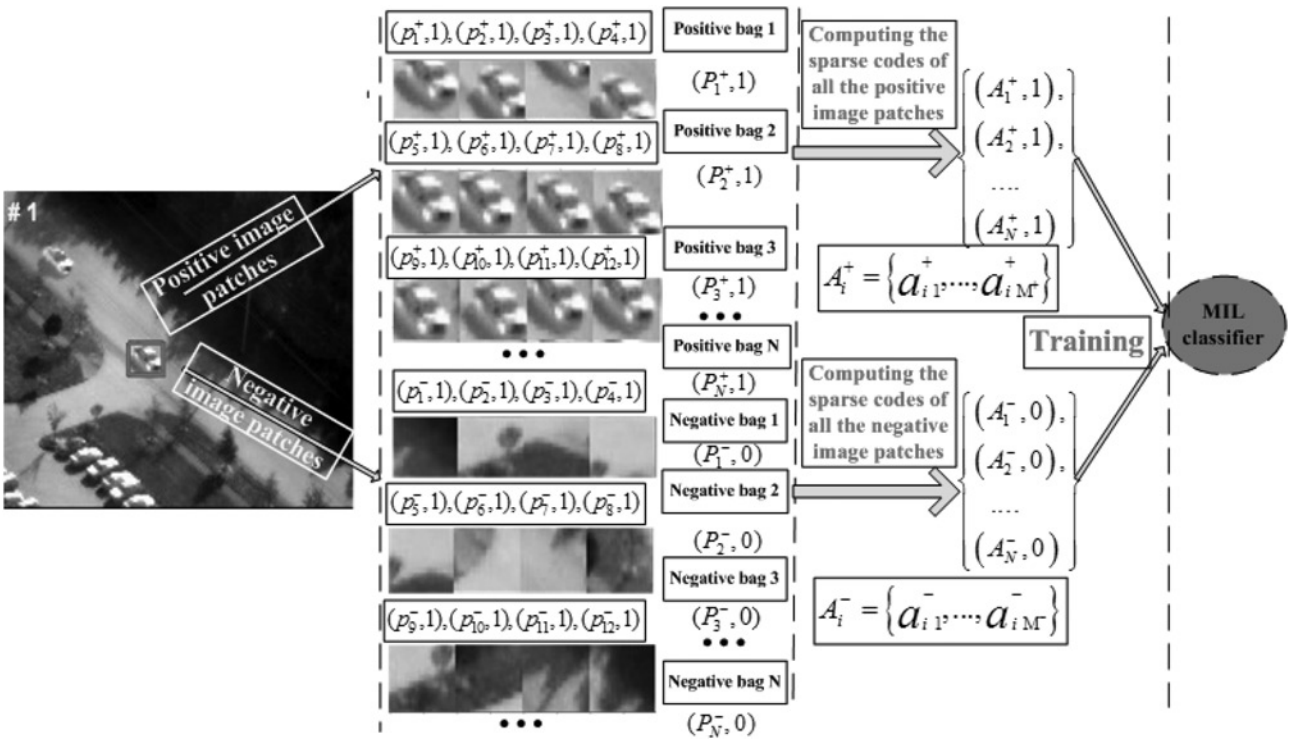

Fig. 1 *A discriminative MIL classifier based on local sparse representation*

To initialise the MIL classifier in the first frame, positive and negative patches around the labelled target are first drawn, and then the corresponding sparse codes of these patches around the target and those from the background are, respectively, computed using a dictionary. Lastly, the MIL classifier is obtained by training the sparse codes of the positive and the negative bags, where A_i^+ and A_i^- correspond to the mapped features of a positive and a negative bag based on sparse codes, respectively, while $a_{iM^-}^-$ and $a_{iM^+}^+$ are the corresponding sparse codes of image patches and computed by (5), respectively.

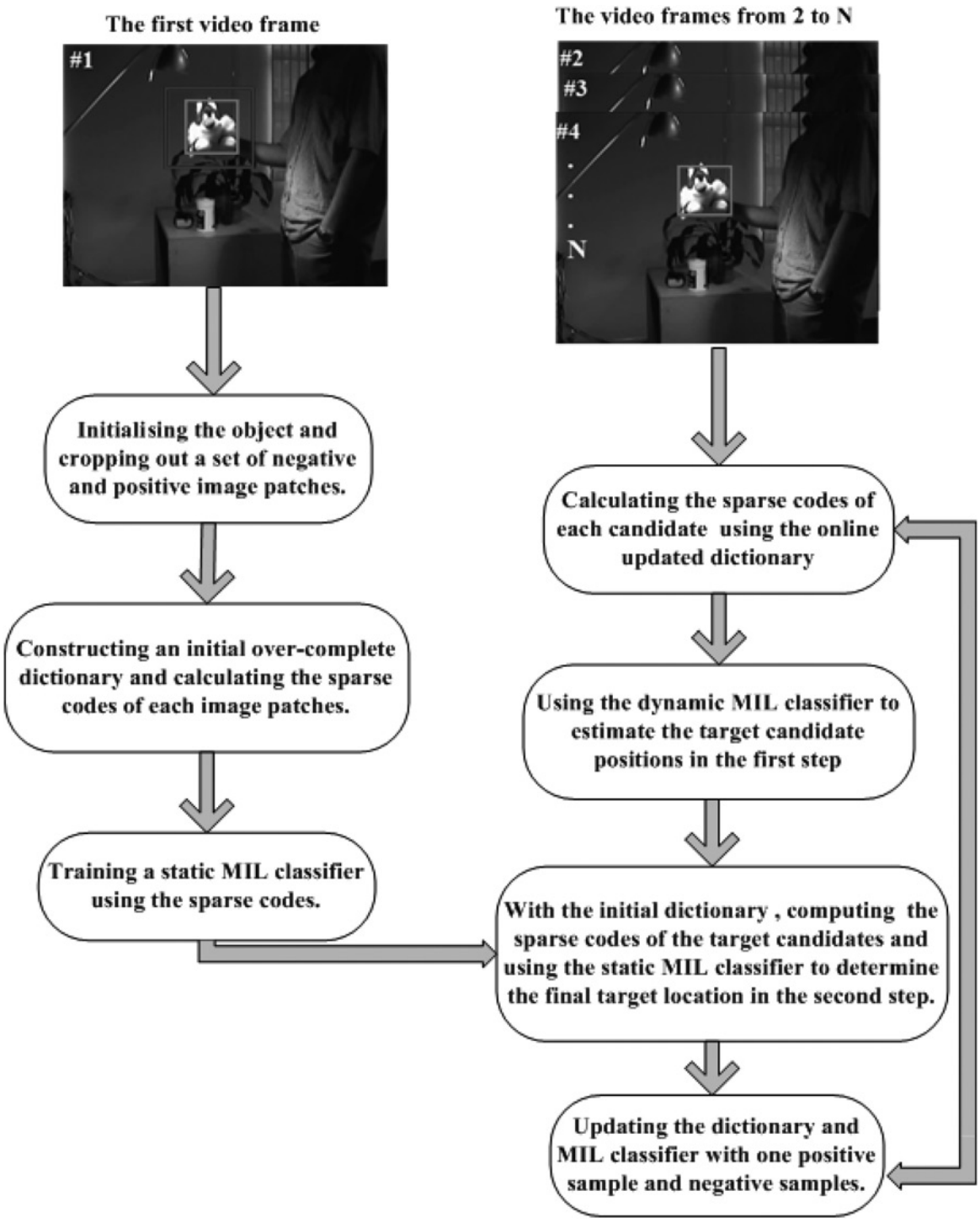

Fig. 2 *Flowchart of the proposed tracking method*

patches, our MIL classifier provides a more flexible mechanism to deal with the problem mentioned above in object tracking. The second contribution is the proposal of a two-step tracking strategy involving the collaboration between the dynamic and static MIL classifiers based on local sparse representation to reduce the drift during tracking process. Different from recent two-step tracking methods which used multiple image features or linear classifier [6, 7], our algorithm incorporates local sparse codes into the framework of the MIL classifier. So the dynamic and static MIL classifier have the desired property that described above and can effectively separate the object from the background in the first step and the second step tracking process.

The remainder of this paper is organised as follows: the related works in appearance models are reviewed in Section 2; Section 3 shows the proposed appearance model based on local sparse representation within the framework of MIL; Section 4 presents a two-step object tracking system with a static and a dynamical MIL classifiers; Section 5 shows our tracking algorithm with particle filter; in Section 6 qualitative and quantitative results of our tracker on six challenging video clips are reported; Some conclusions are made in Section 7.

doi: 10.1049/iet-cvi.2012.0228

2 Related works

Many works have focused on object representation which is a key part of object tracking. A good object representation should have strong description or discrimination power to distinguish targets from background. In general, most algorithms modelled the object appearance by extracting features from global object region [8–11]. Colour histogram was one of the most widely used features [12] and has been implemented in different tracking algorithms [13–17]. However, those trackers didn't work well when objects undergo illumination change or scale change. To address these issues, object representations based on scale-invariant feature transform (SIFT) [18], histogram of oriented gradient (HOG) [19], local binary pattern (LBP) [20] descriptors were designed and recently, a class of appearance modelling techniques named sparse representation [21–23] have been successfully proposed. These works indicated a novel path for solving the problems in the case of object occlusion, and have been successfully applied in robust face recognition [24]. Motivated by the work, more and more work was adopted sparse representation model for tracking objects [2, 7, 25–27]. In [2], each target candidate was represented as a linear combination of a set of online updated templates consisting of target templates and the trivial templates. The candidate with the smallest error for target template reconstruction was obtained as the tracking result. Moreover, Han *et al.* [26] explored an alternative formulation of appearance model with sparse representation, which casted the tracking to find a sparse representation of sub-image feature sets sampled around the target. Experimental results demonstrated the robustness of these approaches; however, the large computational cost prohibited the further application of sparse representation in practice. To accelerate object tracking, a two stage sparse optimisation [28] was proposed by straightforwardly reducing data dimension. More recently, Bai *et al.* [27] presented a structured sparse appearance model for tracking, and block orthogonal matching pursuit was adopted to solve the structured sparse representation problem for reducing the computational cost. But, since the templates that are directly cropped from target image are very limited, the above trackers [27, 28] may fail because the linear representation of the target may not be accurate.

Many of the recent studies have shown that training a model via a discriminative classifier can often performed well in discriminating the object from the background [29–33]. Avidan [29] trained a support vector machine (SVM) classifier offline and applied its extension in an optimal flow framework for object tracking. Furthermore, Avidan [30, 34] developed an online boosting method for tracking targets, which was an ensemble tracker that constructed a strong classifier by training a set of weak classifiers. Zhou *et al.* [35] trained *l*-norm SVMs in a feature space for robust tracking. Grabber and Bischof [36] utilised online AdaBoost algorithm with a proposed novel feature selection method. Parag *et al.* [37] also applied boosting method for tracking, but the form of the weak classifier themselves was updated with the change of background. Grabner *et al.* [38] proposed a semi-supervised approach, where the labelled instances were sampled from the first frame only and the subsequent training samples were left unlabelled. Moreover, different from the above discriminative tracking methods that use a pool of features or a set of boosted classifiers, Wang *et al.* [7] proposed a new algorithm combining sparse codes and a linear classifier directly from raw image patches for object tracking. One important issue the above discriminative tracking methods faced is how to effectively choose positive and negative samples when updating classifiers with newly input samples, since they cannot determine whether these samples are noisy or not.

To address the issue, Babenko *et al.* [4, 5] used MIL instead of traditional supervised learning to handle ambiguous binary data obtained online. In some applications, the MIL tracker can effectively solve the drift problem caused by self-learning in tracking process. Although MIL achieved good results, the selection of the instances in bags was not effective because of existing noisy instances. Once the noisy instances have been used to account for ambiguities in labelling positive instances, the MIL classifier will be degraded when updating itself. Therefore in this paper, we propose a robust method combining online MIL and local sparse representation to select more effective instances for classification.

3 MIL object tracking with the local sparse representation

3.1 Local sparse representation based appearance model

Sparse representations have attracted a great deal of attention in signal processing and have been widely used in many fields including visual tracking [2, 7, 26, 27]. Consider a signal $y \in R^n$, which will be represented as a linear combination of basic elements from a dictionary $D \in R^{n \times K}$ composed of atoms $\{d_j\}_{j=1}^{K}$. A representation of the signal y based on the dictionary $\boldsymbol{D}$ is any vector $x \in R^K$ that satisfies [23]

$$y = \boldsymbol{D}x + z \tag{1}$$

where the dictionary D is said to be overcomplete if $n < K$, and z is a noise term with bounded energy $\|z\|_2 < \varepsilon$. However, the solution of x is generally non-sparse with many non-zero elements. In order to find out a linear combination of only a few atoms to approximate the signal y, this problem can be formally described by Wright *et al.* [24]

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

where $\|\cdot\|_0$ is the l_0 norm that counts the number of non-zero elements, $\|\cdot\|_2$ is the l_2 norm and the parameter ε demonstrates the level of reconstruction error. Since the combinatorial l_0-norm minimisation is an non-deterministic polynomial-time hard (NP-hard) problem, l_1-norm minimisation [2] is applied to replace l_0-norm minimisation and formulated as

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

In our algorithm, a local sparse representation is used to model the appearance of target patches and the corresponding sparse codes are collected to represent the object. Given an image T in the first frame, a set of image patches $\boldsymbol{D} = \{d_i | i = 1{:}K\}$ inside the target region is obtained by sliding a fixed size window, where $\boldsymbol{d}_i \in R^n$ is the ith column for representing a vectorised image patch, n is the dimensionality of image patches and K is the number of image patches. Owing to having overlapped image

patches, similar to Mei and Ling [2], the overcomplete dictionary is constructed by

$$\Phi = [\boldsymbol{D}, \boldsymbol{E}] \tag{4}$$

where $\boldsymbol{E} = [I, -I] \in R^{n \times 2n}$ represents a trivial basis set and maintains a non-negativity constraint of the target coefficient vector, and $\boldsymbol{I} = [i_1, i_2, ..., i_n] \in R^{n \times n}$ is an identity matrix. $\boldsymbol{i}_n \in R^n$ is a vector with one non-zero entry.

Let $\boldsymbol{P} = \{p_i | i = 1{:}M\}$ denote the vectorised image patches extracted from an object image, where $p_i \in R^n$ is the ith image patch, and M is the number of positive image patches and negative image patches. With the dictionary Φ, each p_i will have a corresponding vector of reconstruction coefficients $\boldsymbol{\alpha}_i \in R^{(K+2n)}$, which is computed by

$$\hat{\boldsymbol{\alpha}}_i = \arg\min \|\boldsymbol{\alpha}_i\|_1 \text{ subject to } \|p_i - \Phi\boldsymbol{\alpha}_i\|_2 < \varepsilon \tag{5}$$

When the sparse codes $A = [\alpha_1, \ldots, \alpha_M]$ of all the image patches in an object region are computed, they are collected to represent the object and used to train the MIL classifier for visual tracking.

3.2 MIL

Traditionally, training a binary classifier, that estimates $p(y|x)$, requires a training data set with form $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where each x_i, $i = 1, \ldots, n$ is an instance (sparse codes computed for an image patch in this work), and $y_i \in \{0, 1\}$ is a binary label. Different from the traditional methods, MIL [38] was a generalisation of supervised classification, in which training class label was associated with a set of bags, instead of with individual patterns. Therefore in the multiple instances, learning the training data is formed as $\{(X_1, y_1), \ldots, (X_n, y_n)\}$, where $X_i = \{x_{i1}, \ldots, x_{im}\}$, $i = 1 \ldots. n$ is a bag of instances and y_i is the bag label. The basic idea of MIL is that a bag is assigned as positive label if it contains at least one positive instance, otherwise the bag is negative.

3.3 MIL based on local sparse representation

On the one hand, local sparse representation has been shown to be more robust than others when objects undergo partial occlusion. On the other hand, MIL can handle ambiguities in training data more effectively than other classifiers. Therefore different from classifiers using raw image features, this work integrates the sparse codes from target patches into MIL classifier and discriminates the target from the background in a more robust way.

3.3.1 Sample set based on local sparse representation: Consider a_{ij}^+ is the jth instance in the ith positive bag B_i^+ and the number of positive bags is denoted as i^+, and similarly, a_{ij}^- represents the jth instance in the ith negative bag B_i^- and the number of negative bags is denoted as i^-. Thus $A^+ = \{(A_1^+, 1), \ldots, (A_N^+, 1)\}$ and $A^- = \{(A_1^-, 0), \ldots, (A_N^-, 0)\}$ are used to form the training data for MIL classifier, where

$$A_i^- = S(B_i^-, a_k) = \max \exp(-\|a_{iM^+}^- - a_k\|/\sigma^2) \quad \text{and}$$
$$A_i^+ = S(B_i^+, a_k) = \max \exp(-\|a_{iM^+}^- - a_k\|/\sigma^2)$$

corresponds to the mapped feature of a bag based on sparse codes, whereas $B_i^- = \{a_{i1}^-, \ldots, a_{iM^-}^-\}$ and $B_i^+ = \{a_{i1}^+, \ldots, a_{iM^+}^+\}$ are, respectively, computed by (5). The sample construction of our proposed method is demonstrated in Fig. 1.

3.3.2 MIL classifier with local sparse representation: Numerous algorithms have been proposed for solving the MIL problem [35, 39–41]. Among them, literature [35] trained l-norm SVMs using HOG [19] features within MIL framework for tracking targets, which is similar to our work in this paper. However in this paper, the MIL classifier is trained with local sparse codes from positive and negative bags instead of using features such as intensity, colour, texture and Haar-like features. The l-norm SVMs can be formulated as follows

$$\begin{aligned} \min_{w,b,\xi,\eta} \quad & \lambda \sum_{k=1}^{M} |w_k| + C_1 \sum\nolimits_{i \in B^+} \xi_i + C_2 \sum\nolimits_{j \in B^-} \eta_j \\ \text{s.t.} \quad & (wA_i^+ + b) + \xi_i \geq 1, \quad \forall i \in B^+, \\ & B^+ = \{1, \ldots, i^+\}, \quad \xi_i \geq 0 \\ & -(wA_i^- + b) + \eta_j \geq 1, \quad \forall j \in B^-, \\ & B^- = \{1, \ldots, i^-\}, \quad \eta_j \geq 0 \end{aligned} \tag{6}$$

where ξ_i and η_j are slack variables for positive and negative bags, respectively, and C_1 and C_2 are, respectively, penalty weights for false positives and false negatives. Let w^* and b^* be the optimal solution of (6), where the magnitude of w^* determines the influence of the kth feature on the classifier. Since most elements of the w^* are zero, the index set for non-zero entries in w^* is

$$\Omega = \{k : |w^*| > 0\} \tag{7}$$

The classification of bag B_i is computed as

$$y = \text{sign}\left(\sum_{k \in \Omega} w_k^* S(\boldsymbol{\alpha}_k, B_i) + b^*\right) \tag{8}$$

Equation (8) assigns a label for a bag. An instance $\boldsymbol{\alpha}_{ij}$ in bag B_i is assigned to the positive class, if its contribution to $\sum_{k \in \Omega} w_k^* S(\boldsymbol{\alpha}_k, B_i)$ in (8) is greater than or equal to a threshold, otherwise the negative class is assigned. The threshold is a tradeoff between false positive and false negative classes. Here, the threshold is set to 0.5 and its range is 0–1. The false positive class may be assigned to an instance if the threshold is less than 0.5 therefore the tracking performance is affected by the threshold. To handle ambiguous data, a minimal set of support instances are searched out for efficient visual tracking. Thus, the basic idea of Chen *et al.* [41] is adopted to classify instances based on a bag of classifiers. To reselect the most important instances, an index set is defined as follows

$$\Lambda = \left\{ j^* : j^* = \arg\max_j \exp\left(-\frac{\|\boldsymbol{\alpha}_{ij} - \boldsymbol{\alpha}_k\|^2}{\sigma^2}\right), \; k \in \Omega, \; \boldsymbol{\alpha}_{ij} \in A_i \right\} \tag{9}$$

doi: 10.1049/iet-cvi.2012.0228

In this sense, Λ defines a minimal set of instances responsible for the classification of B_i. Hence, removing the instances $\boldsymbol{\alpha}_{ij}(j^* \notin \Lambda)$ from the bag B_i will not affect the value of $\sum_{k\in\Omega} w_k^* S(\boldsymbol{\alpha}_k, B_i)$ in (8). Since there may have more than one instance in the bag B_i, for each ($j^* \in \Lambda$), a smaller set of instances are defined as follows

$$\Omega_{j*} = \left\{ k{:}k \in \Omega,\ j^* = \arg\max_j \exp\left(-\frac{\left\| \boldsymbol{\alpha}_{ij} - \boldsymbol{\alpha}_k \right\|^2}{\sigma^2} \right) \right\} \tag{10}$$

To efficiently find out the most important support instances, support instances are reselected by (10) and the classification score of a support instance a_s is computed by

$$h(a_s) = \sum_{k\in\Omega_{j*}} \frac{w_k^* S(a_k, a_s)}{m_k} \tag{11}$$

where m_k is denoted as the number of maximisers for a_k.

Once the MIL classifier is initialised, the classification score can be utilised as similarity measure for tracking object. Therefore the larger the classification score of a new support instance is, the more likely the instance is generated from the target class. Subsequently, the instance with the maximum score is considered as the tracking result in current video frame.

4 Two-step object tracking with a static and dynamical MIL classifier

As we know, the ground truth plays a key role in determining whether a new tracking result is effective during tracking process. Since there is no ground truth available in practical applications, noise inevitably occurs in positive instances when updating the observation model involving the dictionary and classifier in this work. Thus, it leads to degrade the discriminability of a classifier in separating targets from the background. To solve the problem, the second novel idea of our proposed algorithm is the use of a dynamic and a static MIL classifier instead of traditional learning to handle ambiguous binary data obtained online. The former one is trained by sparse codes of image patches inside the object region and is updated online for discriminating the target from the background and estimating its candidate locations; the latter one is trained by sparse codes of image patches with ground truth from the first video frame and is used to discriminate the target again from the above same background and to determine its final location.

In the proposed two-step tracking strategy, the first stage captures very large appearance changes and creates a number of candidate positions by a dynamic MIL classifier, while the second stage selects the best candidate and obtains the final results by a static MIL classifier. For different tracking scenarios, the only ground truth is the region of labelled target image in the first frame. So, we first construct a static observation model involving static dictionary Φ_1 and static MIL classifier with parameters w_1^* based on the ground truth. At time t, the initial tracking result is estimated firstly using the online updated dictionary Φ_{t-1} and the dynamic MIL classifier in the first stage. In the second stage, we firstly use the static dictionary Φ_1 to compute the sparse codes of the estimated tracking result. With the static dictionary Φ_1, each image patch p_i of the estimated tracking result will have a corresponding vector of reconstruction coefficients $\hat{\boldsymbol{\alpha}}_i$, which is computed by

$$\hat{\boldsymbol{\alpha}}_i = \arg\min \left\| \boldsymbol{\alpha}_i \right\|_1 \text{ subject to } \left\| p_i - \Phi_1 \boldsymbol{\alpha}_i \right\|_2 < \varepsilon \tag{12}$$

where $\hat{\boldsymbol{\alpha}}_i^{\mathrm{T}} = [\beta_i, \boldsymbol{e}^+, \boldsymbol{e}^-] \in R^{(K+2n)}$ is a non-negative coefficient vector, $\boldsymbol{e}^+, \boldsymbol{e}^- \in R^n$ are called positive and negative trivial coefficient vectors, respectively, in [2]; $\Phi_1 = [\boldsymbol{D}_1, E]$ represents the static dictionary, a set of image patches $\boldsymbol{D}_1 = \{d_i | i = 1{:}K\}$ inside the target region is obtained by sliding a fixed size window in the first frame, and $\boldsymbol{d}_i \in R^n$ is the ith column for representing a vectorised image patch; $\boldsymbol{E} = [\boldsymbol{I}, -\boldsymbol{I}] \in R^{n \times 2n}$ represents a trivial basis set and maintains a non-negativity constraint of the target coefficient vector, and $\boldsymbol{I} = [i_1, i_2, \ldots, i_n] \in R^{n \times n}$ is an identity matrix; $\boldsymbol{i}_n \in R^n$ is a vector and has only one non-zero element, 1. In many visual tracking scenarios, targets are often corrupted by noise or partially occluded and target appearance are also changed. The occlusion and target appearance change create unpredictable errors. Although the tracked target appearance is changed, each image patch p_i of the estimated tracking result from the first stage can still be approximated by $p_i \simeq \Phi_1 \hat{\boldsymbol{\alpha}}_i = \boldsymbol{D}_1 \beta_i + \boldsymbol{e}$ in the second stage. Since $\boldsymbol{e} = [\boldsymbol{e}^+, \boldsymbol{e}^-] \in R^{2n}$ is the error vector and indicates the pixels in the image patch p_i that were corrupted or occluded, the reconstruction coefficients $\hat{\boldsymbol{\alpha}}_i$ can represent the target features effectively even if the feature distance of the target between two frames is too large. Therefore the sparse codes of the initial tracking result from the first step can be computed by (12) with the static dictionary Φ_1 and then they are collected to represent the initial tracking result. Finally, the sparse codes $\hat{\boldsymbol{\alpha}}_i$ of the initial tracking result are collected and used to compute the classification score by the static MIL classifier with parameters w_1^* and, the final tracking result can be determined after the use of particle filter technique.

5 MIL tracking with particle filter

Particle filter [42, 43] 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, the MIL classifier is embedded into the particle filter framework based on sparse representation appearance model. Given the observations of the target $z_{1:t} = \{z_1, \ldots, z_t\}$ up to time t, the current target state $\boldsymbol{s}_t$ can be estimated by maximising a posterior that associates with the highest likelihood

$$\boldsymbol{s}_t = \arg\max_{\boldsymbol{s}_t} p(\boldsymbol{s}_t | z_{1:t}) \tag{13}$$

where $p(\boldsymbol{s}_t | z_{1:t})$ is posterior probability and is recursively computed as

$$p(\boldsymbol{s}_t | z_{1:t}) \propto p(z_t | \boldsymbol{s}_t) \int_{s_{t-1}} p(\boldsymbol{s}_t | \boldsymbol{s}_{t-1}) p(\boldsymbol{s}_{t-1} | z_{1:t-1})\, \mathrm{d}s_{t-1} \tag{14}$$

where $p(z_t | \boldsymbol{s}_t)$ is the observation model or likelihood function that estimates the likelihood of a state given an observation and $p(\boldsymbol{s}_t | \boldsymbol{s}_{t-1})$ is the motion model that predicts the current state, given the previous state.

In this paper, similar to Mei and Ling [2], an affine image warping is applied to model the target motion of two consecutive frames. Let $\boldsymbol{s}_t = (\mu_1, \mu_2, \mu_3, \mu_4, t_1, t_2)$ be the six-dimensional parameter vector for affine transformation, where μ_1, μ_2, μ_3 and μ_4 represent the rotation angle, scale, aspect ratio and skew direction at time t, respectively, and t_1, t_2 are the two-dimensional (2D) position parameters. The transformation of each parameter is represented independently by a scalar Gaussian distribution around their previous state $\boldsymbol{s}_{t-1}$. Then, the motion model is obtained by a Gaussian distribution as follows

$$p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) = N(\boldsymbol{s}_t;\ \boldsymbol{s}_{t-1},\ \boldsymbol{Z}) \tag{15}$$

where $N(\cdot)$ is the Gaussian distribution and $\boldsymbol{Z}$ is the covariance matrix, whose elements are the corresponding variances of affine parameters. The motion model can help generate the candidate samples and save computation cost; however, it may not have fundamental impact on the tracking performance. For the observation model $p(z_t|\boldsymbol{s}_t)$, it can be defined by

$$p(z_t|\boldsymbol{s}_t) \propto h(z_t) \tag{16}$$

where $h(z_t)$ is the MIL classifier defined in (13).

The proposed tracking algorithm is summarised in Fig. 3.

Table 1 Tracking sequences used in our experiments

Video name	Number of frames	Main challenges
faceocc	887	partial occlusion
faceocc2	814	partial occlusion and in-plane pose change
david	462	illumination variation, in-plane/out-of-plane pose change and partial occlusion
sylv	1344	in-plane/out-of-plane pose change, fast motion and illumination change
girl	502	heavy occlusion, fast motion, in-plane/out-of-plane pose change and moving camera
lemming	1336	heavy occlusion, very fast motion and in-plane/out-of-plane pose change

6 Experiments

In this section, we evaluate the performance of our proposed algorithm on six publicly available video sequences involving the challenges of partial or significant occlusion, moving camera, pose and illumination changes, and so on. The details of the selected video sequences are listed in Table 1, where video name, the number of frames and the main challenges are included. For comparison, three state-of-the-art trackers are tested, including the incremental visual tracking tracker (IVT) [44], $L1$ tracking ($L1$ tracker) [2], and MIL tracking tracker [4, 5]. The former two trackers are generative approaches and the latter MIL

1: Input: The initial state of the target $s_1 = (\mu_1, \mu_2, \mu_3, \mu_4, t_1, t_2)$, the video frames $F_1, \ldots F_t$, an initial over-complete dictionary Φ_1 and the learned MIL classifier with parameters w_1^* from the first frame.

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

Initialisation:

3: Initialise the object in the first frame and crop out a set of negative and positive image patches.

4: Construct an initial over-complete dictionary Φ_1 by Equation (4) and compute the sparse codes of each image patches by Equation (5).

5: Train an initial MIL classifier by Equation (6) and obtain classifier parameters w_1^*.

Online tracking:

6: for $t = 2, \ldots, T$ do

7: Sample candidates and calculate the corresponding sparse codes with dictionary Φ_{t-1} by Equation (5).

8: Determine the index set of the most important support instances using Equations (9) and (10).

9: One-step tracker: Estimate the target state s_t' using Equations (11), (12), (14) and (15) with w_{t-1}^*.

10: Two-step tracker: Determine the final tracking result s_t using the dictionary Φ_1 and w_1^* on the basis of the first step result s_t'.

11: Update the MIL classifier parameters w_t^* and the dictionary Φ_t with the final tracking result s_t.

12: End.

Fig. 3 *Algorithm*

doi: 10.1049/iet-cvi.2012.0228

Fig. 4 *Screenshots of tracking results comparison of the proposed tracker (white box) with the L1 tracker (dark grey box), the IVT tracker (black box) and the MIL tracker (grey box), highlighting instances of partial occlusion, illumination variation, heavy occlusion, fast motion, in-plane/out-of-plane pose change and moving camera*

a faceocc
b faceocc2
c david
d sylv
e girl
f lemming

tracker is a discriminative one. For fair comparison, all of them use the same dynamic model and the same particles (600 particles per frame in this work), and they use the MATLAB or C++ codes with the same initialised target locations in these video sequences. These trackers are implemented in MATLAB, except the MIL tracker in C++. The tracking videos, MATLAB or C++ codes, and data sets in [2, 5, 44] can be found from URLs [45–48], respectively.

For qualitative analysis, some representative frames are selected to show the evaluation comparison of our proposed tracker and the others. The performance evaluation can be found in Fig. 4.

6.1 Qualitative analysis

In the six video sequences, the size of a sampled image patch is 16×20. Like other object tracking methods using particle filters [2, 8, 26, 27, 44], the tradeoff between the values of affine parameters should be set and decide how well the values of affine parameters approximate the posterior. In this work, the ranges of the initial values of affine parameters $\boldsymbol{s}_1 = (\mu_1, \mu_2, \mu_3, \mu_4, t_1, t_2)$ are, respectively, 1–10, 1–10, 0.005–0.05, 0.005–0.05, 0.001–0.005 and 0.001–0.005, where μ_1, μ_2, μ_3 and μ_4 represent the rotation angle, scale, aspect ratio and skew direction, respectively, and t_1, t_2 are the 2D position parameters. Therefore to make

Fig. 4 *Continued*

the parameters values suitable, the initial values of t_1, $t_2$will be set to be larger when the target location change was very large between two consecutive frames, such as the sequence 'lemming' and vice verse. The initial of μ_1 will be set to a larger value when the tracked targets encountered greater rotation between two consecutive frames, such as the sequence 'lemming', 'david' and 'girl' and vice verse. The initial values of μ_1, μ_2 will be smaller when the tracked targets underwent smaller scale and aspect ratio change during tracking process and vice verse. Similarly, the initial value of μ_4 is smaller when the tracked targets underwent smaller skew direction change during tracking process and vice verse. The first sequence 'faceocc' involves target objects undergoing partial occlusion and the variances of the affine parameters $\boldsymbol{s}_1$ are set to (2, 2, 0.005, 0.01, 0.001 and 0.001) in this experiment. The second sequence 'faceocc2' is used to test these four tracking methods when target objects undergo partial occlusion and in-plane pose change, and the variances of $\boldsymbol{s}_1$ are set to (2, 2, 0.005, 0.01, 0.001 and 0.001). We utilise the third sequence 'david' to evaluate the trackers with the target undergoing illumination variation, in-plane/out-of-plane pose change and partial occlusion, while the variances of $\boldsymbol{s}_1$ are set to (4, 4, 0.01, 0.02, 0.002 and 0.001). The fourth sequence 'sylv' is used to evaluate the four trackers when the target object undergoes in-plane/out-of-plane pose change, fast motion and illumination change, while the variances of $\boldsymbol{s}_1$ are set to (1, 1, 0.005, 0.005, 0.001 and 0.001). The fifth sequence 'girl' is for the case with the target object undergoing heavy occlusion, fast motion, in-plane/out-of-plane pose change and moving camera, and the variances of $\boldsymbol{s}_1$ are (4, 4, 0.01, 0.01, 0.001 and 0.001). The last sequence 'lemming' is for the case when the target object undergoes heavy occlusion, very fast motion and in-plane/out-of-plane pose change, and the variances of the affine parameters $\boldsymbol{s}_1$ are set to (8, 8, 0.03, 0.02, 0.002 and 0.001) in this experiment.

doi: 10.1049/iet-cvi.2012.0228

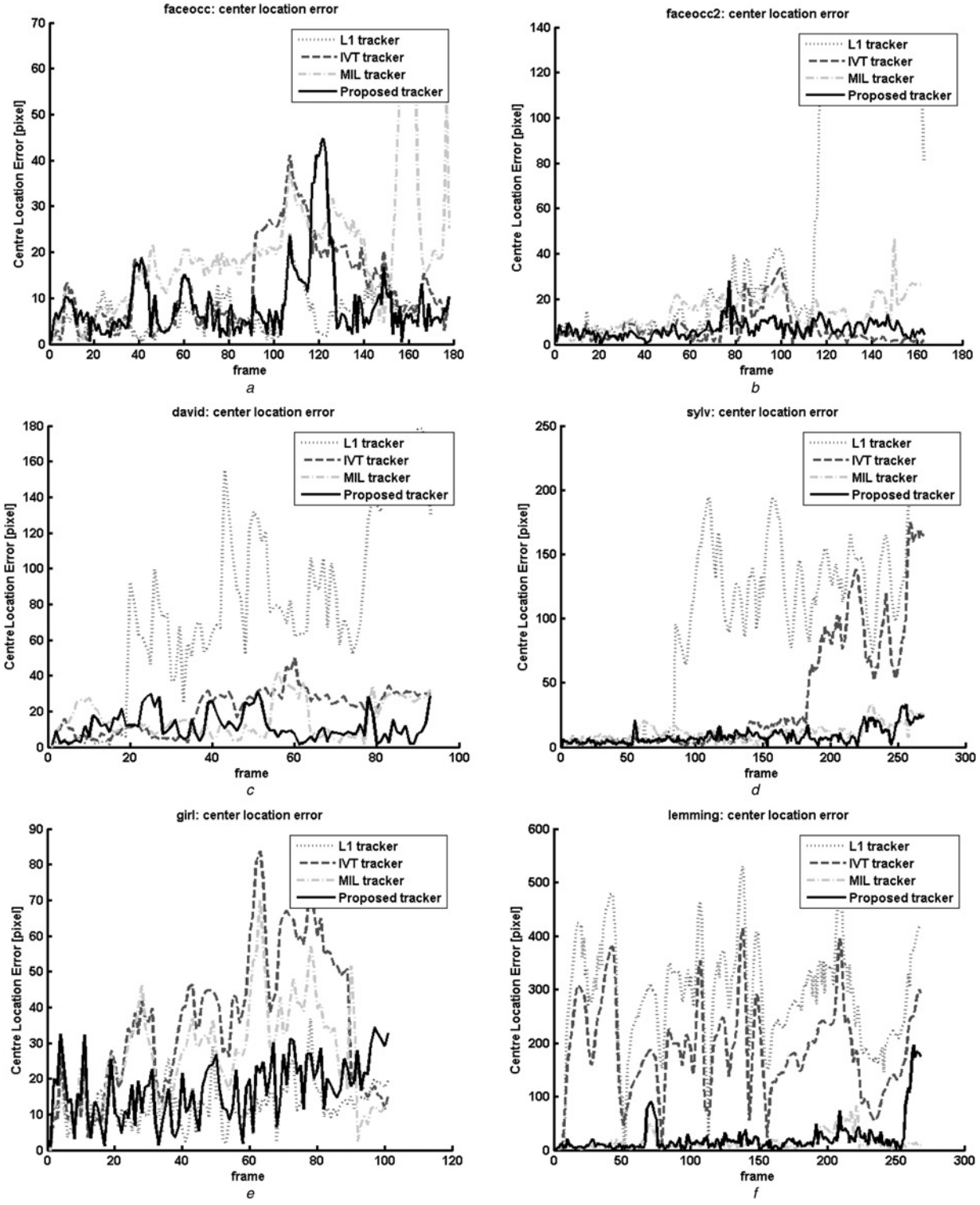

Fig. 5 *Centre location error plots for the proposed tracker, L1 tracker, IVT tracker and MIL tracker*

a faceocc
b faceocc2
c david
d sylv
e girl
f lemming

As seen in Fig. 4*a*, our tracker illustrates competitive performance for the whole sequence frames compared with $L1$ and IVT trackers. However, the target starts to drift in frames 662, 687, 779 and 878 by the MIL tracker. As illustrated in Fig. 4*b*, our tracker and the IVT tracker both can track the face of the man successfully for the whole sequence, while the MIL tracker appears to start target drifting in frame 805 and the $L1$ tracker totally loses the target from the beginning of the frame 494. From Fig. 4*c*, the $L1$ tracker loses the target from the early frames of the sequence because of sudden illumination variation. The MIL tracker and IVT tracker cannot track the target

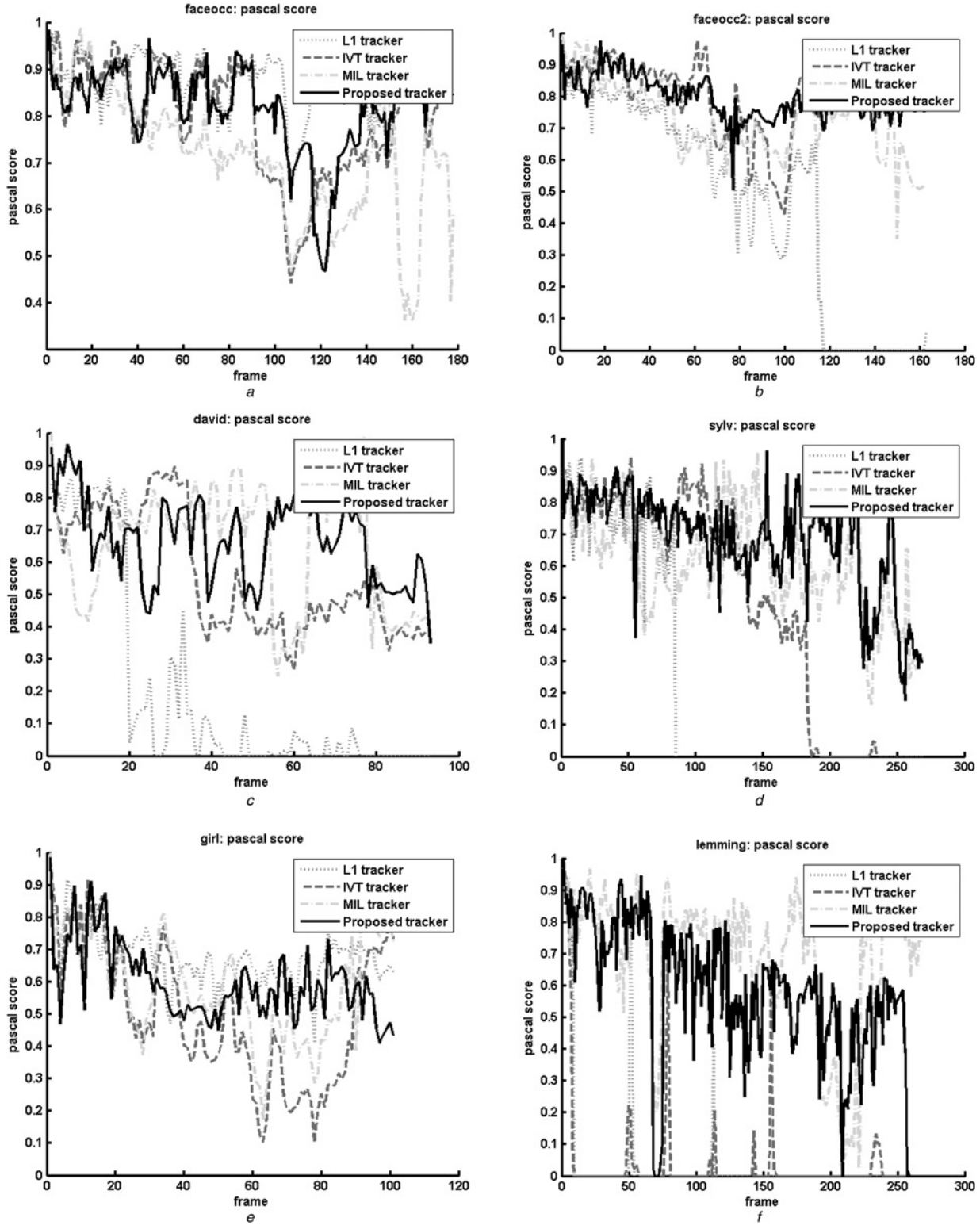

Fig. 6 *Pascal score plots for the proposed tracker, L1 tracker, IVT tracker and MIL tracker*

a faceocc
b faceocc2
c david
d sylv
e girl
f lemming

successfully and drift from the target area in frames 310, 396, 426 and 458. In Fig. 4*d*, the *L*1 tracker fails to track the target after frame 420 and the IVT tracker also misses the target in frame 932. However, both our tracker and the MIL tracker can make satisfied tracking. As shown in Fig. 4*e*, our tracker yields the best performance during the whole sequence. The *L*1 tracker performs the second best partly because it is specifically designed to handle occlusions via sparse approximation with trivial bases, whereas the other two trackers drift away from the target area in frames 138, 213,

doi: 10.1049/iet-cvi.2012.0228

Table 2 Centre location errors (pixel) of the proposed tracker, IVT tracker, *L*1 tracker and MIL tracker

Video name	*L*1 tracker	IVT tracker	MIL tracker	Proposed tracker
girl	**12.992**	35.463	26.520	*16.943*
sylv	91.987	36.417	*11.106*	**8.614**
faceocc	**6.527**	12.472	19.557	*8.690*
david	75.383	20.529	*15.634*	**11.350**
faceocc2	45.406	*6.961*	14.329	**6.680**
lemming	201.917	104.166	**14.891**	*22.082*
overall centre location errors	434.212	216.008	*102.037*	**74.359**

Bold font indicates the best performance; italic font indicates the second best

Table 3 Pascal scores of the proposed tracker, IVT tracker, *L*1 tracker and MIL tracker demonstrate the success rate of the successfully tracked frames for each sequence

Video	*L*1 tracker	IVT tracker	MIL tracker	Proposed tracker
girl	**0.980**	0.446	0.683	*0.782*
sylv	0.309	0.520	*0.751*	**0.851**
faceocc	**1.000**	0.978	*0.993*	0.983
david	0.204	0.462	*0.710*	**0.882**
faceocc2	0.577	0.975	*0.994*	**1.000**
lemming	0.237	0.334	**0.836**	*0.698*
overall Pascal score	3.307	3.715	*4.967*	**5.196**

Bold font indicates the best performance; italic font indicates the second best.

322 and 434. From Fig. 4*f*, it can be seen that the *L*1 tracker and IVT tracker drift away from the target area very quickly because of the very fast motioning of the target. Our tracker and the MIL tracker can track the target well in the whole sequence.

6.2 Quantitative analysis

We use two criteria to evaluate the performance of the proposed tracker quantitatively. The first one is the centre location error that measures the Euclidean distance between the central position of the tracking result and that of the manually labelled ground truth. In our experiments, the ground truth centres of the objects in faceocc, faceocc2, david, sylv, girl and lemming video clips for every five frames are provided by Babenko *et al.* [5] and Santner *et al.* [49]. The second one is the success rate that indicates the number of successful tracked frames. To calculate the success rate, similar to Everingham *et al.* [50], a Pascal score is defined as

$$\text{Pascal score} = \frac{B_R \cap B_T}{B_R \cup B_T} \tag{17}$$

where B_R and B_T are the tracked bounding box and the ground truth bounding box, respectively. Figs. 5 and 6 plot centre location error and Pascal score, respectively, for our tracker, *L*1 tracker, IVT tracker and MIL tracker. The centre location errors and Pascal scores are reported in Tables 2 and 3, respectively. From experimental results, we can see that our proposed tracking algorithm outperforms the others. As shown in Table 2, our method achieves the best performance on the 'sylv', 'david' and 'faceocc2' sequences. Although *L*1 tracker and MIL tracker perform better than our tracker in the cases of 'girl', 'faceocc' and 'lemming' video clips, our method has the lowest overall centre location errors implying that it is more stable than the other three trackers. From Pascal scores in Table 3, we also observe that our tracker has the highest success rate compared with the other three trackers, except for the cases of the 'girl', 'faceocc' and 'lemming' sequences. Considering overall Pascal score, our tracker has a score 5.196, which indicates that it is of the highest success rate than the other trackers in the six experiments.

6.3 Two-step tracking analysis

To illustrate the process of the two-step object tracking, a case of experiment on the toy video sequence is shown in Fig. 7. Since the accumulative errors and no ground truth available when updating the dictionary and the classifier, using only dynamic MIL classifier with dynamic dictionary results in slight drift from the target in frame 576. However, in the same frame, using the two-step strategy involving static MIL classifier and static dictionary with the ground truth of the first frame can effectively alleviate the drift problem during the tracking process.

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 dynamic MIL classifier and repeated the evaluation on the sequences, girl, sylv, faceocc, david and faceocc2.

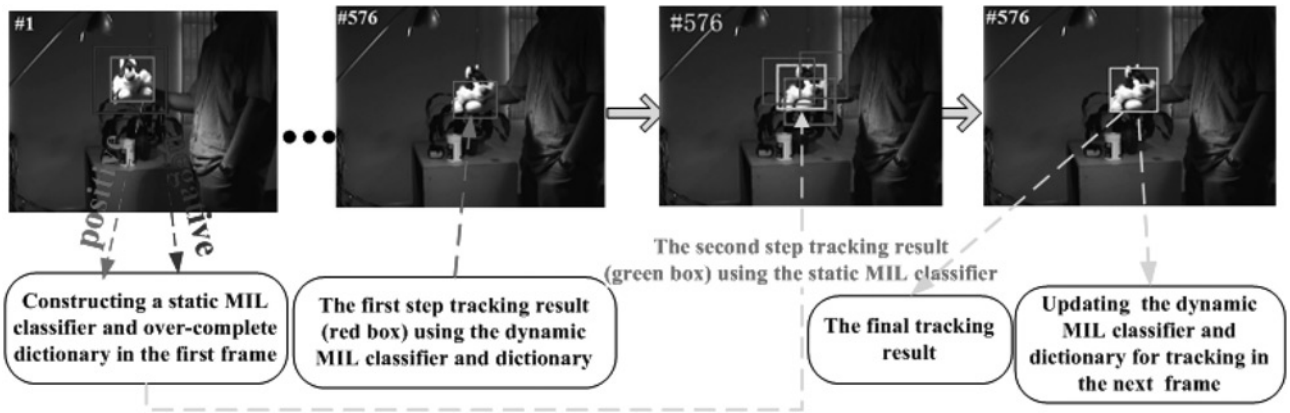

Fig. 7 *Process of two-step object tracking*

Fig. 8 *Screenshots of tracking results comparison of the one-step tracker (dark grey box) with the two-step tracker (grey box) on the girl, sylv, faceocc, david and faceocc2 video sequences, respectively*

a faceocc
b faceocc2
c david
d sylv
e girl

doi: 10.1049/iet-cvi.2012.0228

Fig. 8*a* illustrates the final tracking results on the eight representative video frames, where the two-step tracker shows competitive performance in the whole sequence frames, whereas the one-step tracker starts target drifting after frame 89. As illustrated in Fig. 8*b*, both the two-step tracker and one-step tracker can obtain competitive results in the first 496 frames; however, the latter one starts to make the target drift away from frame 531. From Fig. 8*c*, we find that the one-step tracker makes the target drift away from frame 220 and finally loses the target after frame 244, because it does not have the ability to objectively capture the pose and illumination changes, while the two-step tracker with the ground truth of the first frame can. In Fig. 8*d*, the two-step tracker can track the face of the man during the entire sequence, whereas the one-step tracker is unable to obtain satisfied results from frame 823 and fails to locate the target after frame 846. As shown in Fig. 8*e*, the two-step tracker achieves a good performance during the tracking process, and the one-step tracker drifts away from the target site in frames 322 and 357 and further fails to track the target from frame 404.

7 Conclusion

When an object undergoes significant pose change, illumination variation and/or partial occlusion, the representation of objects plays a very important role in the robustly and adaptively visual object tracking. This paper proposes a novel approach with the design of appearance model, that is, online MIL classifier based on local sparse codes. Different from traditional classifier using raw image features, this work instead adopts the sparse codes of local image patches with an overcomplete dictionary for object representation, and uses a MIL classifier to discriminate the target object from the background. The learned MIL classifier is then embedded into a Bayesian inference framework to construct a robust tracking algorithm. In addition, to alleviate the influence of the drift problem when updating the proposed tracker, we put forward a two-step tracking strategy with a static and dynamical classifier. With this tracking strategy, experiments on some challenging video sequences show that the proposed tracker achieves state-of-the-art performance in qualitative and quantitative respects under partial occlusion, illumination, pose variation and so on.

8 Acknowledgment

This work was supported by the NSFC-Guangdong Joint Foundation Key Project under grant (no. U1135003), the National Nature Science Foundation of China (grant no. 61070227).

9 References

1 Yilmaz, A., Javed, O., Shah, M.: 'Object tracking: a survey', *ACM Comput. Surv.*, 2006, **38**, (4), pp. 1–45
2 Mei, X., Ling, H.: 'Robust visual tracking and vehicle classification via sparse representation', *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 2011, **33**, (11), pp. 2259–2272
3 Dietterich, T.G., Lathrop, R.H., Perez, L.T.: 'Solving the multiple-instance problem with axis parallel rectangles', *Artif. Intell.*, 1997, **88**, (1–2), pp. 31–71
4 Babenko, B., Yang, M.-H., Belongie, S.: 'Visual tracking with online multiple instance learning'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June 2009, pp. 983–990
5 Babenko, B., Ming-Hsuan, Y., Belongie, S.: 'Robust object tracking with online multiple instance learning', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (8), pp. 1619–1632
6 Matthews, I., Ishikawa, T., Baker, S.: 'The template update problem', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004, **26**, (6), pp. 810–815
7 Wang, Q., Chen, F., Xu, W., Yang, M.-H.: 'Online discriminative object tracking with local sparse representation'. WACV '12 Proc. 2012 IEEE Workshop on the Applications of Computer Vision, January 2012, pp. 425–432
8 Wang, J., Chen, X., Gao, W.: 'Online selecting discriminative tracking features using particle filter'. Proc. IEEE Conf. on CVPR, June 2005, pp. 1037–1042
9 Han, Z.J., Ye, Q.X., Jiao, J.B.: 'Online feature evaluation for object tracking using kalman filter'. 19th Int. Conf. on Pattern Recognition, December 2008, pp. 1–4
10 Han, Z.J., Ye, Q.X., Jiao, J.B.: 'Feature evaluation by particle filter for adaptive object tracking'. Proc. SPIE Visual Communication and Image Processing, 2009
11 Wang, J.Q., Yagi, Y.S.: 'Integrating color and shape-texture features for adaptive real-time object tracking', *IEEE Trans. Image Process.*, 2008, **17**, (2), pp. 235–240
12 Collins, R.T., Liu, Y., Leordeanu, M.: 'Online selection of discriminative tracking features', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, **27**, (10), pp. 1631–1643
13 Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: 'Color-based probabilistic tracking'. Proc. European Conf. on Computer Vision, 2002, pp. 661–675
14 Comaniciu, D., Ramesh, V., Meer, P.: 'Kernel-based object tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (5), pp. 564–575
15 Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: 'Robust online appearance models for visual tracking', *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 2003, **25**, (10), pp. 1296–1311
16 Ning, J., Zhang, L., Zhang, D., Wu, C.: 'Scale and orientation adaptive mean shift tracking', *IET Comput. Vis.*, 2012, **6**, (1), pp. 52–61
17 Ning, J., Zhang, L., Zhang, D., Wu, C.: 'Robust mean shift tracking with corrected background-weighted histogram', *IET Comput. Vis.*, 2012, **6**, (1), pp. 62–69
18 Lowe, D.: 'Distinctive image features from scale-invariant key points', *Int. J. Comput. Vis.*, 2004, **60**, (2), pp. 91–110
19 Dalal, N., Triggs, B.: 'Histograms of oriented gradients for human detection'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June 2005, pp. 886–893
20 Ojala, T., Pietikainen, M., Maenpaa, T.: 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (7), pp. 971–987
21 Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., Yan, S.: 'Sparse representation for computer vision and pattern recognition', *Proc. IEEE*, 2010, **98**, (6), pp. 1031–1044
22 Candès, E., Romberg, J., Tao, T.: 'Stable signal recovery from incomplete and inaccurate measurements', *Commun. Pure Appl. Math.*, 2006, **59**, (8), pp. 1207–1223
23 Donoho, D.: 'Compressed sensing', *IEEE Trans. Inf. Theory*, 2006, **52**, (4), pp. 1289–1306
24 Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: 'Robust face recognition via sparse representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (2), pp. 210–227
25 Lin, H.X., Shen, C.H., Shi, Q.F.: 'Real-time visual tracking using compressive sensing'. IEEE Conf. on CVPR, June 2011, pp. 1305–1312
26 Han, Z., Jiao, J., Zhang, B., Ye, Q., Liu, J.: 'Visual object tracking via sample-based adaptive sparse representation (AdaSR)', *Pattern Recognit.*, 2011, **44**, (9), pp. 2170–2183
27 Bai, T., Li, Y.F.: 'Robust visual tracking with structured sparse representation appearance model', *Pattern Recognit.*, 2012, **45**, (6), pp. 2390–2404
28 Liu, B., Yang, L., Huang, J., Meer, P., Gong, L., Kulikowski, C.: 'Robust and fast collaborative tracking with two stage sparse optimization'. Proc. ECCV, 2010, no. 4, pp. 624–637
29 Avidan, S.: 'Support vector tracking', *IEEE Trans. Pattern Anal. Mach. Intell.* , 2004, **26**, (8), pp. 1064–1072
30 Avidan, S.: 'Ensemble tracking', *IEEE Trans. Pattern Anal. Mach. Intell.* , 2007, **29**, (2), pp. 261–271
31 Collins, R.T., Liu, Y.: 'On-line selection of discriminative tracking features'. Proc. IEEE Conf. on Computer Vision, June 2003, pp. 346–352
32 Kalal, Z., Matas, J., Mikolajczyk, K.: 'P-N learning: bootstrapping binary classifiers by structural constraints'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2010, pp. 49–56
33 Yu, Q., Dinh, T.B., Medioni, G.: 'Online tracking and reacquisition using co-trained generative and discriminative trackers'. Proc. European Conf. on Computer Vision, 2008, pp. 678–691
34 Avidan, S.: 'Ensemble tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, **29**, (2), pp. 261–271
35 Zhou, Q.H., Lu, H.C., Yang, M.-H.: 'Online multiple support instance tracking'. Proc. IEEE Conf. Automatic Face and Gesture Recognition, March 2011, pp. 545–552

36 Grabner, H., Bischof, H.: 'On-line boosting and vision'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June 2006, pp. 260–267
37 Parag, T., Porikli, F., Elgammal, A.: 'Boosting adaptive linear weak classifiers for online learning and tracking' (CVPR, 2008), pp. 1–8
38 Grabner, H., Leistner, C., Bischof, H.: 'Semi-supervised on-line boosting for robust tracking'. Proc. European Conf. on Computer Vision, 2008, pp. 234–247
39 Andrews, S., Tsochantaridis, I., Hofmann, T.: 'Support vector machines for multiple-instance learning'. Proc. NIPS, 2002, pp. 561–568
40 Viola, P., Platt, J.C., Zhang, C.: 'Multiple instance boosting for object detection' (NIPS, 2007), pp. 1417–1426
41 Chen, Y., Bi, J., Wang, J.Z.: 'Miles: multiple-instance learning via embedded instance selection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (12), pp. 1931–1947
42 Doucet, A., Freitas, N.de., Gordon, N.: 'Sequential monte carlo methods in practice' (Springer, 2001)
43 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.*, 2002, **50**, (2), pp. 174–188
44 Ross, D., Lim, J., Lin, R.S., Yang, M.-H.: 'Incremental learning for robust visual tracking', *Int. J. Comput. Vis.*, 2008, **77**, (1), pp. 125–141
45 http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml
46 http://www.dabi.temple.edu/~hbling/code_data.htm
47 http://www.cs.toronto.edu/~dross/ivt/
48 http://gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php
49 Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: 'Prost: parallel robust online simple tracking'. June 2010, pp. 723–730
50 Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: 'The pascal visual object classes (voc) challenge', *Int. J. Comput. Vis.*, 2010, **88**, (2), pp. 303–338

doi: 10.1049/iet-cvi.2012.0228