Robust Visual Tracking Using Dynamic Classifier Selection with Sparse Representation of Label Noise

Yuefeng Chen and Qing Wang

School of Computer Science and Engineering
Northwestern Polytechnical University, Xi'an 710072, P.R. China

Abstract. Recently a category of tracking methods based on "tracking-by-detection" is widely used in visual tracking problem. Most of these methods update the classifier online using the samples generated by the tracker to handle the appearance changes. However, the self-updating scheme makes these methods suffer from drifting problem because of the incorrect labels of weak classifiers in training samples. In this paper, we split the class labels into true labels and noise labels and model them by sparse representation. A novel dynamic classifier selection method, robust to noisy training data, is proposed. Moreover, we apply the proposed classifier selection algorithm to visual tracking by integrating a part based online boosting framework. We have evaluated our proposed method on 12 challenging sequences involving severe occlusions, significant illumination changes and large pose variations. Both the qualitative and quantitative evaluations demonstrate that our approach tracks objects accurately and robustly and outperforms state-of-the-art trackers.

1 Introduction

Visual object tracking has been one of the most attractive topics in computer vision and there are numerous practical applications so far, such as motion analysis, video surveillance, traffic controlling and so on. In recent years, although many tracking methods [4,21,22,3,28,15,10,16] have been proposed and made a certain breakthrough, it is still a challenging issue to design a tracking algorithm which is robust to severe occlusions, significant illumination changes, pose variations, fast motion, scale changes and background clutter.

Generally, a typical tracking system contains three basic components: object representation, appearance model and motion model. The appearance model is used to represent the object and provide prediction accordingly. The motion model (such as Kalman filter [6], Particle filter [21,22,27]) is applied to predict the motion distribution of the object between two adjacent frames. In this paper, we focus on the appearance model which is the most important and challenging part of the tracker.

In literatures [4,21,2,26], most of tracking approaches can be categorized as either generative or discriminative ones based on different appearance models.

K.M. Lee et al. (Eds.): ACCV 2012, Part III, LNCS 7726, pp. 29–42, 2013.

The generative methods formulate the tracking problem as locating the object region with *maximum* probability generated from the **modeled** appearance model [22,6]. Adam *et al.* [1] proposed a robust fragments-based tracking approach where the object is represented by multiple image fragments. This tracker can handle partial occlusions and pose changes well. Since the template was fixed and not updated during tracking, the tracker is sensitive to appearance changes of the object. To acclimate to appearance changes, the template or appearance model of the tracker should be updated dynamically. Ross *et al.* [22] developed a tracking algorithm based on incrementally learning a low-dimensional subspace representation. Kwon *et al.* [15] decomposed the observation model into multiple basic observation models and the proposed method was robust to significant motion and appearance changes.

For discriminative methods, the tracking problem is formulated as training a discriminative classifier to separate the object from surrounding background. These methods are also noted as tracking-by-detection methods since training a classifier is similar to detection problem. Recently, numerous tracking approaches based on detection are proposed [4,8,9,23,11,17,13]. In order to handle appearance changes, online ensemble learning was popularly applied to the tracking problem. In [2], Avidan proposed an online ensemble tracking algorithm by online learning. Grabner *et al.* [8] proposed an online Adaboost feature selection method. Babenko *et al.* [4] used online MIL (Multiple Instance Learning) instead of traditional supervised learning to handle the drifting problem. In their work, the training samples are constructed as bags and labels are corresponded to bags rather than samples. Classifier selection scheme plays an important role in online ensemble learning based tracking methods. In [8], classifiers are selected by measuring the accumulated error. In [4], by maximizing the log likelihood of bags, classifiers are selected from the classifier pool. However, these classifier selection algorithms almost assume the training dataset is clean, which do not meet practical applications of visual tracking. Noisy training dataset will reduce the performance of the classifier and cause drifting during tracking.

Recently, sparse representation [26] has been widely applied in object tracking [21,28,20,19,18,12]. Mei *et al.* [21] treated tracking as a sparse approximation problem in a particle filter framework. Each target candidate is sparsely represented in the space of object templates and trivial templates, and the candidate with the smallest projection error is taken as the target. This representation is robust to partial occlusions and other changes. However it is computationally expensive to obtain sparse coefficients by ℓ_1 *minimization*. This restriction makes it unsuitable for real-time applications. Bao *et al.* [5] developed a very fast numerical solver to tackle the ℓ_1 *minimization* problem based on the accelerated proximal gradient (APG) algorithm such that the tracker can work in real time. Zhang *et al.* [28] expanded the ℓ_1 tracker by employing popular sparsity-inducing $\ell_{p,q}$ mixed norms and formulated object tracking in a particle filter framework as a multi-task sparse learning problem. Most of these methods use sparse coding to represent the appearance model of the object. However, little work has been done on sparse representation for the class labels of classifiers.

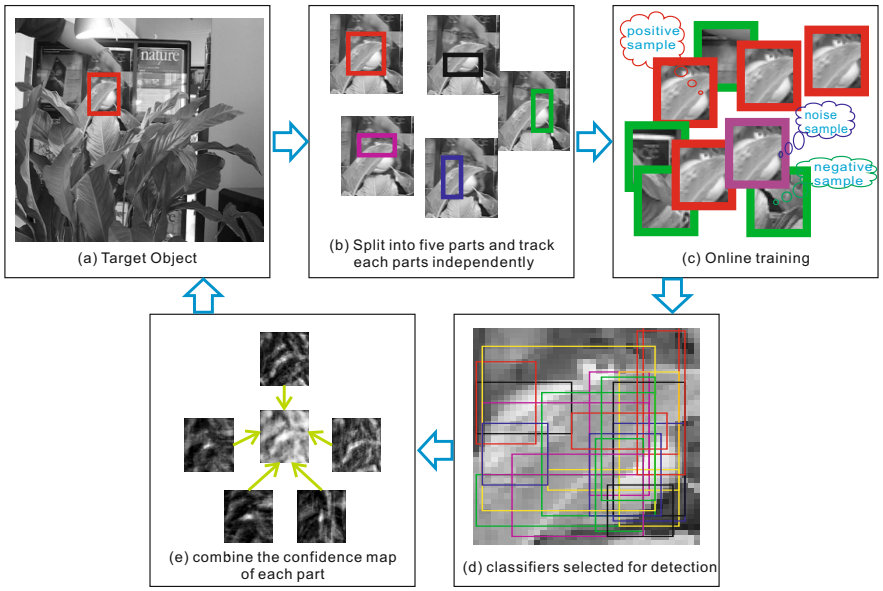

Fig. 1. The pipeline of the proposed object tracking approach. In (c), the sample with purple box is a noisy one since it is actually positive but labeled as negative.

In this paper, we present a novel dynamic classifier selection with sparse representation of label noise. We use sparse representation to model the class labels and obtain classifiers by ℓ_1 minimization. In order to verify the robustness of the selection scheme, the algorithm is integrated into a boosting framework to solve visual tracking problem. In order to handle severe occlusion issue, we expand the traditional boosting algorithm to a part based one which is more robust to occlusion. Figure 1 depicts an overview of the proposed tracking approach. The key contributions of our work can be concluded as follows.

- We split the class labels into true labels and noisy labels and model them using sparse representation. Moreover, we propose a novel dynamic classifier selection algorithm based on ℓ_1 *minimization*, which is more robust than traditional selection schemes in [4,8].
- We expand online boosting framework and employ the proposed dynamic classifier selection to this framework. And then a more robust tracking approach is proposed and achieves better performance comparing with other methods.

The rest of the paper is organized as follows. Dynamic classifier selection is proposed in section 2. In section 3, we integrate dynamic classifier selection and an expanded online boosting framework to object tracking. Experimental results are shown and analyzed in section 4 with qualitative and quantitative evaluations. The concluding remarks are drawn in section 5.

2 Dynamic Classifier Selection with Label Noise

In this section, we describe the proposed dynamic classifier selection algorithm with class label noise in details. First we discuss the motivation of our work. Then, the model of classifier with class label noise is presented. At last, we solve this problem using ℓ_1 minimization.

2.1 Motivation

In the field of machine learning, ensemble learning methods has been popular in recent years and achieved better performance than traditional learning methods. One of the most critical tasks is to select a group of classifier dynamically in the weak classifier pool [14]. Many dynamic classifier selection methods [8,4] have been proposed in recent years, however, most of them do not pay attention to the class label noise which is inherent in the real world dataset. The performance of ensemble learning methods can be rapidly degraded when there are some noises in the training dataset. Therefore, developing a noise-tolerant dynamic classifier selection algorithm is a significant task. In this work, we model the class labels using sparse representation based on the assumption that noise class labels are sparsely distributed over the training dataset. Finally, the weak classifier with the maximum value in the sparse vector will be selected.

2.2 Sparse Representation of Class Labels

Given the training dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$ and label $\mathbf{y} = [y_1, y_2, \ldots, y_L]^\top \in \mathbb{R}^{L\times 1}, y_i \in \{-1, +1\}$. The class label vector predicted by the weak classifier $\mathbf{h}$ in this feature set $\mathbf{X}$ is denoteds as,

$$\mathbf{h}(\mathbf{X}) = [h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_L)]^\top, \tag{1}$$

where $\mathbf{h}(\mathbf{X}) \in \mathbb{R}^{L\times 1}$ and $h(\mathbf{x}_i)$ is the predicted class label of sample $\mathbf{x}_i$ by the weak classifier h. Then the joint class labels corresponding to different weak classifiers are denoted by,

$$\mathbf{\Phi} = [\mathbf{h}_1(\mathbf{X}), \mathbf{h}_2(\mathbf{X}), \ldots, \mathbf{h}_M(\mathbf{X})], \tag{2}$$

where $\mathbf{\Phi} \in \mathbb{R}^{L\times M}, L \ll M$, L is the number of training samples, and M is the number of weak classifiers in a pool which are used for selection. Assuming that the true label vector $\hat{\mathbf{y}}$ without noise is a linear combination of $\mathbf{h}_i(\mathbf{X})$, thus $\hat{\mathbf{y}}$ can be represented as,

$$\hat{\mathbf{y}} \approx \mathbf{\Phi}\boldsymbol{\beta} = \beta_1\mathbf{h}_1(\mathbf{X}) + \beta_2\mathbf{h}_2(\mathbf{X}) + \ldots + \beta_M\mathbf{h}_M(\mathbf{X}), y_i \in \{-1, +1\}, \tag{3}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_M]^\mathsf{T} \in \mathbb{R}^{M\times 1}$ is a coefficient vector corresponding to those M weak classifiers, and the weak classifier with the *maximum* coefficient in $\boldsymbol{\beta}$ will be selected.

However, $\hat{\mathbf{y}}$ can not be obtained ideally in practical applications since the data is not always clean. For example, on visual tracking problem, training data is generated automatically and class labels are obtained by the trained classifier. Since the classifier is not perfectly correct, there will be several misclassified training samples in the training set. In order to solve this problem, we split the class labels $\mathbf{y}$ into true labels $\hat{\mathbf{y}}$ and noise labels $\mathbf{e}$ such that $\hat{\mathbf{y}}$ is described as $\hat{\mathbf{y}} = \mathbf{y} - \mathbf{e}$. Thus Equation (3) is changed to,

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \mathbf{e}. \tag{4}$$

In addition, as the predicted labels must be positively related to $\mathbf{y}$, we add a non-negativity constraint to these coefficients. The final formulation is shown in Equation (5).

$$\mathbf{y} = \begin{bmatrix}\mathbf{\Phi}\ \mathbf{I}\ -\mathbf{I}\end{bmatrix} \begin{bmatrix}\boldsymbol{\beta}\\ \mathbf{e}^{+}\\ \mathbf{e}^{-}\end{bmatrix} = \mathbf{W}\mathbf{c},\ s.t.\quad \mathbf{c} \geq \mathbf{0}, \tag{5}$$

where $\mathbf{W} = \begin{bmatrix}\mathbf{\Phi}\ \mathbf{I}\ -\mathbf{I}\end{bmatrix} \in \mathbb{R}^{L\times(M+2L)}$ and $\mathbf{c} = \begin{bmatrix}\boldsymbol{\beta}\\ \mathbf{e}^{+}\\ \mathbf{e}^{-}\end{bmatrix} \in \mathbb{R}^{(M+2L)\times 1}$ is a non-negative coefficient vector.

2.3 Classifier Selection Using ℓ_1 Minimization

Based on the above formulation, we herein want to have a sparse solution to (5). When we obtain a solution, the $\boldsymbol{\beta}^*$ should be a sparse vector corresponding to weak classifiers and $\mathbf{e}^+$ and $\mathbf{e}^-$ correspond to the class label noises respectively. In our approach, the weak classifier which has the *maximum* value in the sparse vector $\boldsymbol{\beta}^*$ is selected, *i.e.*,

$$h^{sel} = h^{m^*},\ \text{where } m^* = \underset{i}{argmax}\ \ \beta_i^*. \tag{6}$$

The optimal coefficients $\mathbf{c}^*$ can be obtained by solving the following ℓ_0 optimization objective function. However, the ℓ_0 minimization is NP-hard problem, thus ℓ_1 minimization is instead.

$$\mathbf{c}^* = \underset{\mathbf{c}}{argmax}\ \ \|\mathbf{c}\|_1 \quad s.t.\quad \mathbf{y} = \mathbf{W}\mathbf{c},\ \mathbf{c} \geq \mathbf{0} \tag{7}$$

And Equation (7) can be changed into Lagrangian optimization problem and solved in polynomial time [26].

$$\mathbf{c}^* = \underset{\mathbf{c}}{argmax}\ \ \|\mathbf{W}\mathbf{c} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{c}\|_1 \quad s.t.\quad \mathbf{c} \geq \mathbf{0} \tag{8}$$

Figure 2 gives some experiments and intermediate results of our method on *tiger* sequence. Figure 2(d) shows a sample of patches corresponding to those weak classifiers selected by the proposed algorithm. In order to get the spatial distribution of these patches, we map them to the object rectangle, as shown in Figure 2(c).

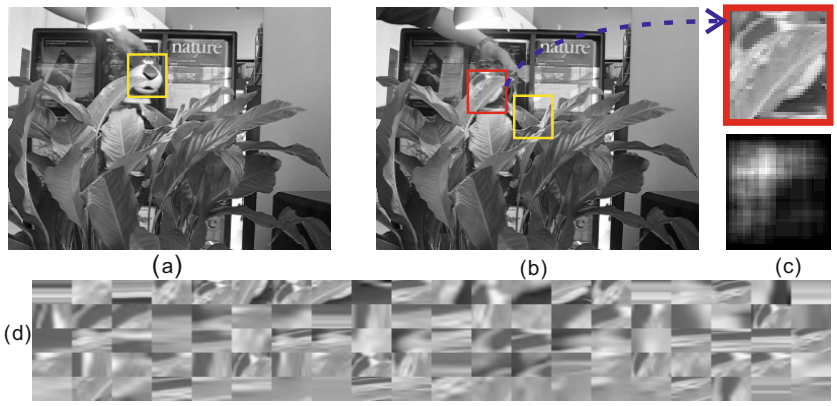

Fig. 2. Tracking the *tiger* with significant pose variations and heavy occlusion. (a) Initial tracking target (yellow rectangle). (b) Tracking results of our method (in red) and expanded online boosting method (in yellow). (c) Spatial distribution of these patches over the red rectangle. (d) Patches corresponding to the classifiers obtained from dynamic classifier selection algorithm. It shows that most of the selected patches are concentrated at the top left part which is not occluded by the tree leaf.

3 Visual Tracking Using Dynamic Classifier Selection

In this section we employ the above mentioned dynamic classifier selection algorithm to an expanded online boosting framework. And then a novel tracking approach which is robust to severe occlusion, fast motion, illumination changes and pose variations is proposed. The basic framework of track-by-detection algorithm is depicted in Figure 3. First we will give a brief introduction of motion model. Then the proposed method will be described in detail.

Generally, almost all the motion models in tracking-by-detection algorithms [4,8,9,23,11,17,13] are based on sampling. The goal of tracking is to choose the sample with the *maximal* confidence $p_s\left(y=1|\mathrm{p}_t\right)$ by an online trained classifier, where p_t represents the location of estimated 2D patch in the search region at frame $f_t \in \mathcal{F}, t=1,\ldots,T$. As shown in Figure 3, the tracker maintains the object location p^*_{t-1} (marked with red fork) at time $t-1$. In order to estimate the object location at time t, a number of patches $\mathrm{P}_t = \left\{\mathrm{p}_t \middle| \left\|\mathrm{p}_t - \mathrm{p}^*_{t-1}\right\|_2^2 \leq r\right\}$ will be generated to predict the location with a radius r. The patch p^*_t with the *maximum* confidence becomes the object location at time t.

$$\mathrm{p}^*_t = \underset{\mathrm{p}_t \in \mathrm{P}_t}{argmax}\ p_s\left(y=1|\mathrm{p}_t\right) \tag{9}$$

This motion model is called translational motion model. Although there exist some more sophisticated motion models, such as particle filter [21,22,27], combined motion model [15], we mainly concentrate on the appearance model during tracking, especially in the scheme of online learning.

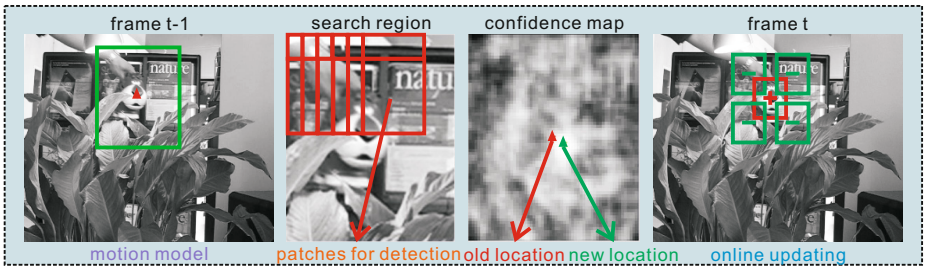

Fig. 3. The basic framework of tracking-by-detection algorithms

Given a patch located at $\mathrm{p} \in \mathrm{P}_t$, let $\mathbf{x}^{\mathrm{p}}$ denote the feature vector extracted from the image patch in the frame f_t. The training samples are formulated as $\langle \mathbf{x}^{\mathrm{p}}, y \rangle, y \in \{-1, +1\}$, where $+1$ and -1 represent the sample belonging to the object or background. The strong classifier, $H(\mathbf{x}^{\mathrm{p}})$, has the following form,

$$H(\mathbf{x}^{\mathrm{p}}) = \sum_{n=1}^{N} \alpha_{n,sel} \cdot h_{n,sel}(\mathbf{x}^{\mathrm{p}}), \tag{10}$$

where $h_{n,sel}(\mathbf{x}^{\mathrm{p}})$ is the n^{th} selected weak classifier and weighted by $\alpha_{n,sel}$. As the standard online boosting [8] mentioned, $\alpha_{n,sel}$ can be expressed as $\frac{1}{2}\ln\frac{1-e_{n,sel}}{e_{n,sel}}$ where $e_{n,sel}$ is the accumulated error of the weak classifier and is formulated as $\frac{\lambda^w_{n,sel}}{\lambda^w_{n,sel}+\lambda^c_{n,sel}}$. $\lambda^c_{n,sel}$ is the number of samples correctly classified and $\lambda^w_{n,sel}$ is the number of samples misclassified.

In order to deal with severe occlusion, we expand online boosting framework by employing part based scheme and propose a part based boosting algorithm in this paper, with the assumption that there will always be a part of the object which is not occluded. The K strong classifiers are combined by adopting the Noisy-OR [25] model. Let $\mathrm{p}_t \circ l_k$ indicate the patch location in the k^{th} part of the object, the model can be written as,

$$p_s(y = 1|\mathrm{p}_t) = 1 - \prod_{k=1}^{K} \left(1 - H^k\left(\mathbf{x}^{\mathrm{p}_t \circ l_k}\right)\right) \tag{11}$$

The algorithm of part based boosting framework and online updating based on dynamic classifier selection are shown in Algorithm 1 and Algorithm 2 respectively.

4 Experiments

In this section, we verify the proposed tracker on 12 challenging tracking sequences (*e.g. animal, coke11, david, dollar, faceocc, faceocc2, girl, shaking, surf, sylv, tiger1, tiger2*). The 1^{st} and 8^{th} sequence can be obtained from [15], and

Algorithm 1. Part based boosting framework

Input: Current image f_t, object location p^*_{t-1} at time $t-1$.
Output: Object location p^*_t at time t.

$\mathrm{P}_t = \left\{\mathrm{p}_t \middle| \|\mathrm{p}_t - \mathrm{p}^*_{t-1}\|_2^2\right\}$;
$p_s\left(y=1|\mathrm{p}_t\right) = 1 - \prod_{k=1}^{K}\left(1 - H^k\left(\mathbf{x}^{\mathrm{p}_t \circ l_k}\right)\right)$;
$\mathrm{p}^*_t = argmax_{\mathrm{p}_t \in \mathrm{P}_t} p_s\left(y=1|\mathrm{p}_t\right)$;
Generate training samples and label them using the combined strong classifier;
Update each strong classifier using Algorithm 2.

Algorithm 2. Dynamic classifier selection embedded online updating

Input: Training samples $\chi = \{\langle \mathrm{p}_1, y_1\rangle, \langle \mathrm{p}_2, y_2\rangle, \ldots, \langle \mathrm{p}_L, y_L\rangle\}$, $y_i \in \{+1, -1\}$
Strong classifiers $H^k(\mathbf{x}), k = 1, 2, \ldots, K$.
Output: Strong classifiers $H^k(\mathbf{x}), k = 1, 2, \ldots, K$.

for $k = 1$ **to** K **do**
  for $n = 1$ **to** N **do**
    for $m = 1$ **to** M **do**
      $h^k_{m,n}\left(\mathbf{x}^{\mathrm{p}_i \circ l_k}\right) = \mathtt{Update}\left(h^k_{m,n}\left(\mathbf{x}^{\mathrm{p}_i \circ l_k}\right), \left\langle \mathbf{x}^{\mathrm{p}_i \circ l_k}, y_i\right\rangle\right)$;
      $\lambda^{k,w}_{n,m} = \lambda^{k,w}_{n,m} + \sum_i \mathbf{1}\left(h^k_{m,n}\left(\mathbf{x}^{\mathrm{p}_i \circ l_k}\right) \neq y_i\right)$;
      $\lambda^{k,c}_{n,m} = \lambda^{k,c}_{n,m} + \sum_i \mathbf{1}\left(h^k_{m,n}\left(\mathbf{x}^{\mathrm{p}_i \circ l_k}\right) = y_i\right)$;
      $e^k_{n,m} = \frac{\lambda^{k,w}_{n,m}}{\lambda^{k,w}_{n,m} + \lambda^{k,c}_{n,m}}$;
    end
    $\mathbf{X} = \left[\mathbf{x}^{\mathrm{p}_1 \circ l_k}, \mathbf{x}^{\mathrm{p}_2 \circ l_k}, \ldots, \mathbf{x}^{\mathrm{p}_L \circ l_k}\right]$, $\mathbf{y} = \left[y_1, y_2, \ldots, y_L\right]^{\mathsf{T}}$;
    $\mathbf{\Phi} = \left[\mathbf{h}^k_{n,1}(\mathbf{X}), \mathbf{h}^k_{n,2}(\mathbf{X}), \ldots, \mathbf{h}^k_{n,M}(\mathbf{X})\right]$, $\mathbf{W} = \left[\mathbf{\Phi}\ \mathbf{I}\ -\mathbf{I}\right]$;
    get $\mathbf{c}^*$ via solving Equation (8);
    $m^* = argmax_i \boldsymbol{\beta}^*$;
    $h^k_{n,sel} = h^k_{n,m^*}, \alpha^k_{n,sel} = \frac{1}{2}\ln\frac{1-e^k_{n,m^*}}{e^k_{n,m^*}}$;
  end
  $H^k = \sum_{n=1}^{N} \alpha^k_{n,sel} \cdot h^k_{n,sel}$;
end

the others are available at [4]. The proposed method is compared to the latest state-of-the-art tracking algorithms, including fragment-based tracker (FRAG) [1], ℓ_1 tracker (ℓ_1) [21], multiple instance learning (MIL) tracker [4], visual tracking decomposition (VTD) approach [15] and the method of part based online boosting without ℓ_1 minimization called ours (w/o ℓ_1) using the same initial position. The source codes of FRAG[1], ℓ_1-tracker[2], MIL[3], VTD[4] can be found at URLs.

[1] `http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm`
[2] `http://www.dabi.temple.edu/~hbling/code_data.htm`
[3] `http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml`
[4] `http://cv.snu.ac.kr/research/~vtd/`

Our proposed algorithm is implemented in MATLAB, and can process about 15 frames per second. The features we used in this paper are the standard Haar-like features [24]. Each weak classifier $h(\mathbf{x})$ corresponds to a Haar-like feature. We use the following weak classifier model,

$$h(\mathbf{x}^{\mathrm{p}}) = \begin{cases} +1 & p_w(y=1|\mathbf{x}^{\mathrm{p}}) \geq p_w(y=-1|\mathbf{x}^{\mathrm{p}}) \\ -1 & otherwise \end{cases} \tag{12}$$

where $p_w(y=1|\mathbf{x}^{\mathrm{p}})$ and $p_w(y=-1|\mathbf{x}^{\mathrm{p}})$ are two Gaussian distributions, *i.e.*, $\mathcal{N}(\mu^+, \sigma^+)$ for positive samples and $\mathcal{N}(\mu^-, \sigma^-)$ for negative samples respectively. In addition, we adopt Kalman filter [6] to update weak classifiers.

Almost all the parameters in the experiments are fixed. We set the number of strong classifiers $K = 5$, corresponding to the raw position, half top, half bottom, half left and half right respectively. The number of weak classifier N is 10 for sequence (*coke11, dollar, faceocc, sylv*) and 20 for others, and the number of weak classifiers M in the pool for selection is 200. The parameter λ in ℓ_1 minimization is fixed to 0.01.

In order to eliminate the effects which are brought by randomness, we run each video sequence 5 times and average the results.

4.1 Qualitative Evaluation

In the sequence *animal*, several deers are running in water very fast. Figure 4(a) shows the tracking results on *animal* sequence. FRAG, MIL tracker, ℓ_1 tracker, ours (w/o ℓ_1) tracker failed at frame 39 because of fast motion. Only ours (w/ ℓ_1) and VTD tracker can successfully track the object throughout this sequence. The result shows the robustness of ours (w/ ℓ_1) algorithm in handling fast motion.

In the sequence *coke11*, a moving coke can is suffering illumination changes, occlusion and pose variations. The tracking results are shown in Figure 4(b). The FRAG, ℓ_1 tracker and VTD tracker failed at frame 50 after illumination variation. The other algorithms provide robust tracking result for this sequence.

In the third sequence *david*, light and pose changes significantly throughout the whole sequence. As shown in Figure 4(c), the pose of David changes heavily between frame 117 and 161. And all the other algorithms drift when the pose varies wildly. This result clearly demonstrates that ours (w/ ℓ_1) can handle pose change robustly.

In the sequence *dollar*, some dollars are folded and then divided into two parts. The appearance is changed when dollars are folded. From Figure 4(d), we can conclude that the FRAG, ℓ_1 tracker, and VTD tracker failed to track the dollars when the appearance changed. Ours (w/ ℓ_1) and (w/o ℓ_1) and MIL can track the dollars through the whole sequence. The reason for the better performance is that these algorithms are online learning based algorithms.

In the sequence *faceocc*, the woman's face is partly occluded by a book. Figure 4(e) shows the tracking results. MIL tracker drifts at frame 250 after the occlusion. VTD tracker fails to track the face between frame 577 and frame 600. The other algorithms can track the occluded face throughout the sequence.

The next the sequence is *faceocc2*, which is more challenging than *faceocc*. Except the occlusion, the man's appearance is changed (wearing a hat) among the sequence. In Figure 4(f), FRAG and VTD trackers failed when the head is rotated. ℓ_1 tracker loses the face when the man wears a hat. Ours (w/o ℓ_1) algorithm tracks the face poorly after frame 707. However, ours (w/ ℓ_1) method is able to track the face throughout the whole sequence.

The results of sequence *girl* are shown in Figure 4(g). In this sequence, occlusion and pose variations are taken place during the sequence. Ours (w/ ℓ_1), ℓ_1 tracker and VTD tracker can successfully track the girl. The other algorithms failed as the pose change and occlusion.

In the sequence *shaking* (Figure 4(h)), the variations of illumination and pose are very severe. The results show that all the other algorithms except ours (w/ ℓ_1) and VTD tracker drift at frame 61 when light is changed significantly. We show the robustness of our algorithm to severe illumination and pose changes.

The challenges of sequence *surfer* are fast motion, pose change and scale change. As shown in Figure 4(i), only FRAG and ℓ_1 tracker lost the object during tracking. The other algorithms can faithfully track the object through the sequence.

In the sequence *sylv*, a moving animal doll is suffering from lighting variations, scale and pose changes. The results are shown in Figure 4(j). ℓ_1 tracker failed after the frame 624 and VTD tracker, FRAG also failed after frame 1179. Only MIL tracker, ours (w/ ℓ_1) and ours (w/o ℓ_1) can track the object through the long sequence.

The results of sequence *tiger1* are shown in Figure 4(k). Many trackers drift at frame 76 because of occlusion and fast motion, pose changes. Only ours (w/ ℓ_1) can deal with these issues well.

In the last sequence *tiger2*, trackers except MIL tracker lost the tiger at frame 113 (Figure 4(l)). Our algorithm can not perform well on this sequence when fast motion and occlusion occurred simultaneously.

4.2 Quantitative Evaluation

For the quantitative comparison, position error and overlap criterion in PASCAL VOC [7] are employed. They are computed as,

$$error\,(R_G, R_T) = \|center\,(R_G) - center\,(R_T)\|_2^2 \tag{13}$$

$$overlap\,(R_G, R_T) = \frac{area\,(R_G \cap R_T)}{area\,(R_G \cup R_T)} \tag{14}$$

Table 1 summarizes the average position error and Table 2 shows the success rate of tracking methods. Figure 5 depicts the average position error of trackers on each tested sequence. Overall, our proposed algorithm achieves best performance on the sequence *david*, *dollar*, *feaceocc2*, *surfer*, *sylv*, *tiger1* against state-of-the-art methods, and on the other sequences its performance is comparable to the best method.

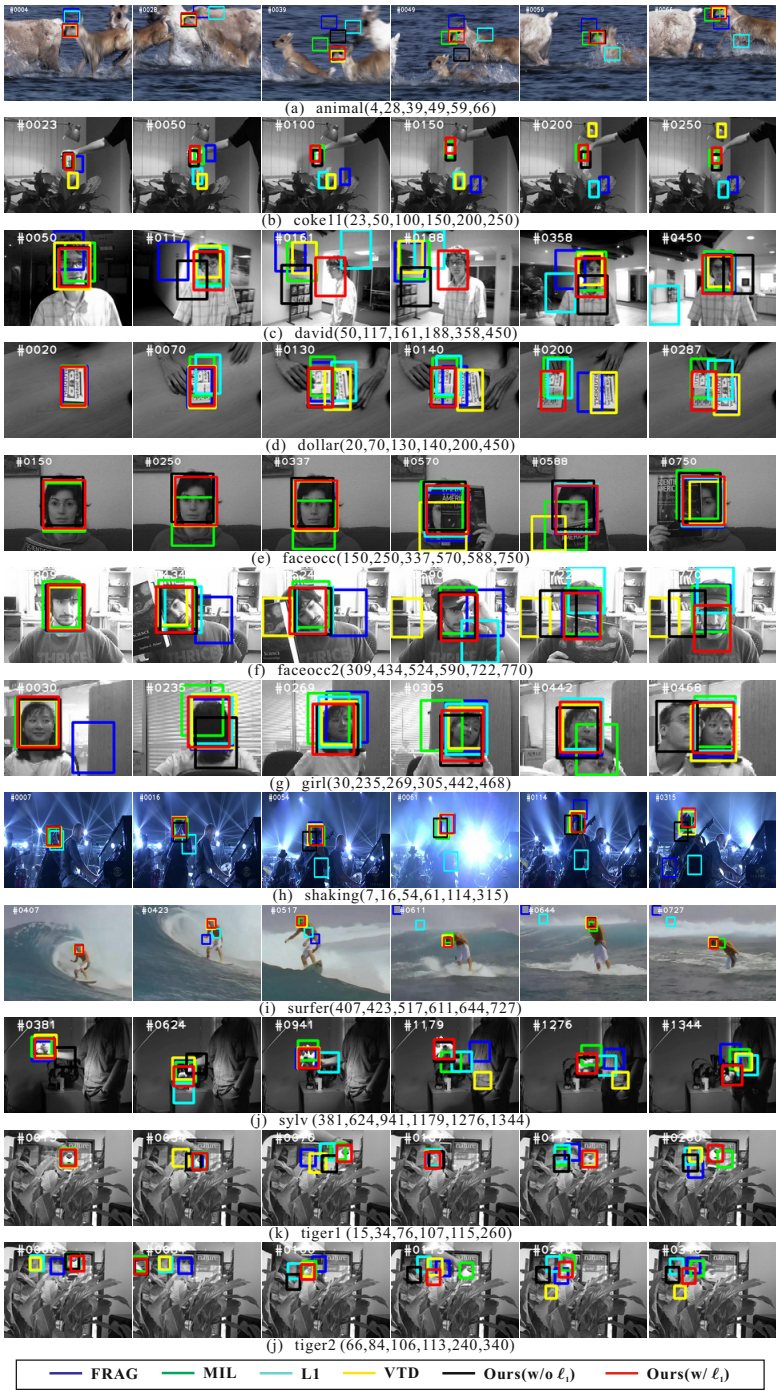

Fig. 4. Tracking results of different algorithms for 12 challenging sequences

Table 1. Average position error of 6 trackers (best: red, second: blue)

	FRAG	MIL	ℓ_1	VTD	Ours(w/o ℓ_1)	Ours(w/ ℓ_1)
animal	86.9	37.3	133.4	**9.7**	23.9	**18.5**
coke11	63.3	25.4	52.7	62.8	**8.8**	**20.0**
david	45.9	**25.4**	78.1	26.5	46.4	**11.5**
dollar	56.2	39.3	27.1	65.7	**5.9**	**5.5**
faceocc	**5.3**	24.9	6.4	11.8	16.6	**5.9**
faceocc2	30.9	22.7	31.2	52.3	**16.4**	**8.5**
girl	23.9	33.6	**13.1**	**13.9**	22.1	14.4
shaking	110.2	32.1	145.1	**5.5**	36.1	**11.7**
surfer	149.1	9.0	114.6	7.9	**6.6**	**5.7**
sylv	21.4	**11.9**	21.5	21.9	14.1	**7.6**
tiger1	39.6	**31.2**	33.6	42.0	34.4	**8.2**
tiger2	37.6	**9.7**	46.8	54.1	53.8	**27.4**
Average	25.2	25.2	58.6	31.2	**23.8**	**12.1**

Table 2. Success rate of 6 trackers (best: red, second: blue)

	FRAG	MIL	ℓ_1	VTD	Ours(w/o ℓ_1)	Ours(w/ ℓ_1)
animal	0.04	0.43	0.03	**0.99**	0.73	**0.78**
coke11	0.08	0.18	0.14	0.07	**0.68**	**0.32**
david	0.48	**0.61**	0.27	0.70	0.44	**0.94**
dollar	0.41	0.68	0.20	0.39	**1.00**	**1.00**
faceocc	**1.00**	0.79	**1.00**	0.91	0.88	**1.00**
faceocc2	0.39	0.84	0.64	0.58	**0.85**	**0.99**
girl	0.83	0.50	**0.99**	0.95	0.89	**0.98**
shaking	0.18	0.59	0.01	**1.00**	0.48	**0.84**
surfer	0.14	0.64	0.08	0.68	**0.89**	**0.92**
sylv	**0.72**	0.66	0.55	0.65	0.64	**0.90**
tiger1	0.20	**0.33**	0.20	0.23	0.29	**0.87**
tiger2	0.12	**0.66**	0.10	0.15	0.15	**0.38**
Average	0.58	0.58	0.35	0.61	**0.66**	**0.83**

5 Conclusion

In this paper, we propose a dynamic classifier selection algorithm and integrate it to an expanded online boosting framework. In the proposed dynamic classifier selection algorithm, we divided the class labels into two parts which are true and noise class labels. Based on the formulation, the classifier selection is solved by ℓ_1 *minimization*. In the proposed online boosting framework, we employ the part-based idea in which object is represented by several patches. Compared to some state-of-the-art trackers on 12 challenging sequences, the experimental results demonstrate that the proposed method is more accurate and more robust to appearance variations, pose changes, occlusions, fast motion and so on.

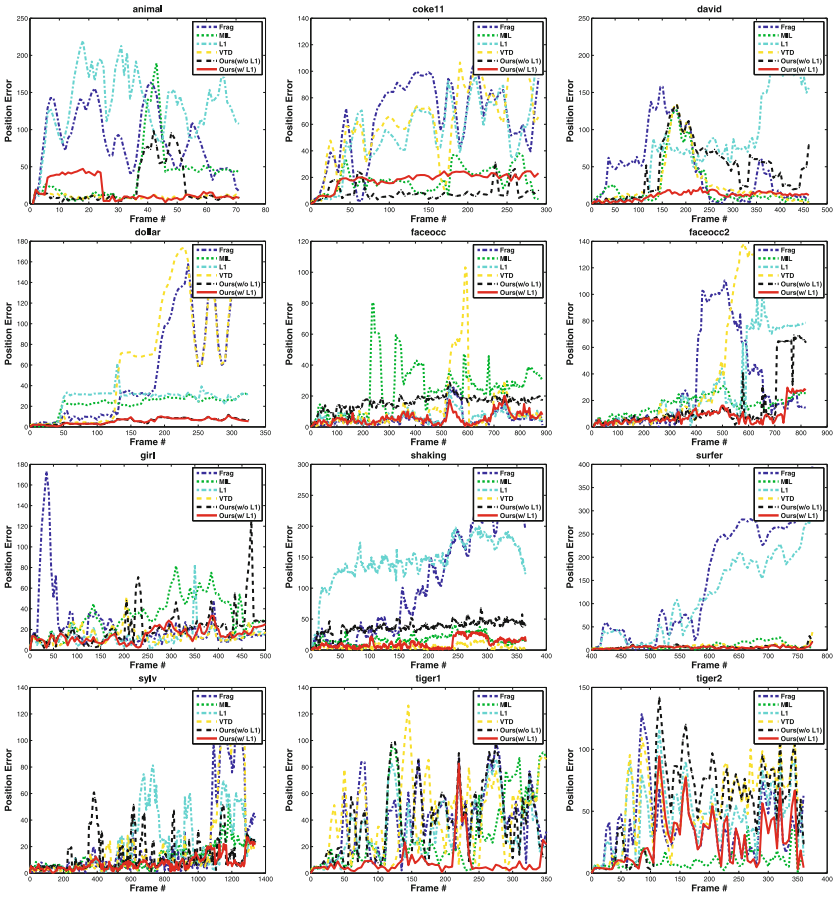

Fig. 5. Quantitative comparisons of average position error (in pixels)

Acknowledgement. This work was supported by NSFC funds (61103060 and 61272287), National "863" Programs under grant (2012AA011803) and graduate starting seed fund of Northwestern Polytechnical University (Z2012135), P. R. China.

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR, vol. (1), pp. 798–805 (2006)
2. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. 29, 261–271 (2007)
3. Bai, T., Li, Y.F.: Robust visual tracking with structured sparse representation appearance model. Pattern Recognition 45, 2390–2404 (2012)
4. Babenko, B., Yang, M.H., Belongie, S.J.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)

5. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust ℓ_1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25, 564–575 (2003)
7. Everingham, M., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88, 303–338 (2010)
8. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR, vol. (1), pp. 260–267 (2006)
9. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised On-Line Boosting for Robust Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
10. Gong, H., Sim, J., Likhachev, M., Shi, J.: Multi-hypothesis motion planning for visual object tracking. In: ICCV, pp. 619–626 (2011)
11. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In: ICCV, pp. 263–270 (2011)
12. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
13. Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: Bootstrapping binary classifiers by structural constraints. In: CVPR, pp. 49–56 (2010)
14. Ko, A.H.R., Sabourin, R., de Souza Britto Jr., A.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognition 41, 1718–1731 (2008)
15. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR, pp. 1269–1276 (2010)
16. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: ICCV, pp. 1195–1202 (2011)
17. Li, G., Qin, L., Huang, Q., Pang, J., Jiang, S.: Treat samples differently: Object tracking with semi-supervised online covboost. In: ICCV, pp. 627–634 (2011)
18. Li, X., Shen, C., Shi, Q., Dick, A., van den Hengel, A.: Non-sparse linear representations for visual tracking with online reservoir metric learning. In: CVPR (2012)
19. Liu, B., Huang, J., Yang, L., Kulikowski, C.A.: Robust tracking using local sparse appearance model and k-selection. In: CVPR, pp. 1313–1320 (2011)
20. Liu, B., Yang, L., Huang, J., Meer, P., Gong, L., Kulikowski, C.: Robust and Fast Collaborative Tracking with Two Stage Sparse Optimization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 624–637. Springer, Heidelberg (2010)
21. Mei, X., Ling, H.: Robust visual tracking using ℓ_1 minimization. In: ICCV, pp. 1436–1443 (2009)
22. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77, 125–141 (2008)
23. Stalder, S., Grabner, H., van Gool, L.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: ICCV Workshops, pp. 1409–1416 (2009)
24. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. (1), pp. 511–518 (2001)
25. Viola, P.A., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2005)
26. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. 31, 210–227 (2009)
27. Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., Cheng, E.: Blurred target tracking by blur-driven tracker. In: ICCV, pp. 1100–1107 (2011)
28. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR (2012)