Visual Tracking with Weighted Online Feature Selection

Yu Tang[1], Zhigang Ling[1], Jiancheng Li[2], and Lu Bai[2]

[1] Electrical and Information Engineering Institution of Hunan University, Changsha, China
[2] China Highway Engineering Consulting Group Co, Ltd, Beijing, China

Abstract. Most tracking-by-detection algorithms adopt an online learning classifier to separate targets from their surrounding background. These methods set a sliding window to extract some candidate samples from the local regions surrounding the former object location at current frame. The trained classifier is then applied to these samples, which sample with the maximum classifier score is considered as the new object location. However, in classifier training procedure, noisy samples may often be included when they are not *correct* enough, thereby causing visual drift. Online discriminative feature selection (ODFS) method has been recently introduced into the tracking algorithms, which can alleviate drift to some extent. However, the ODFS tracker may detect the candidate sample that is less accurate because it does not discriminatively take the sample importance into consideration during the feature selection procedure. In this paper, we present a novel weighted online discriminative feature selection (WODFS) tracker, which integrates the sample's contribution into the optimization procedure when selecting features, the proposed method optimizes the objective function in the steepest ascent direction with respect to the weighted positive samples while in the steepest descent direction with respect to the negative. Therefore, the selected features directly couple their scores with the contribution of samples which result in a more robust and stable tracker. Numerous experiments on challenging sequences demonstrate the superiority of the proposed algorithm.

Keywords: visual tracking, feature selection, online learning, tracking by detection.

1 Introduction

Recently, Visual tracking has become a very hot research topic in the field of computer vision because of its wide applications, e.g. video indexing, traffic monitoring, and human computer interaction [1] etc. Numerous methods have been proposed in the past decades [2-10]. However, it is still a challenging task to develop a robust tracking algorithm that works universally for diverse application, because tracker often suffer from some factors such as appearance changes, pose variations, partial or full

S. Li et al. (Eds.): CCPR 2014, Part I, CCIS 483, pp. 168–182, 2014.

occlusions and illumination changes. Therefore designing a robust appearance model [21] that can adapt to these factors becomes a main task in most recently proposed algorithms [1-9]. According to different appearance model, the recently proposed tracking algorithms can be classified into two classes based on their difference representation scheme: generative models[2,3,4,22]and discriminative models[5,6,7,8,9].

Generative models typically learn an appearance model to represent the target, and then search for the target region with minimal error [20]. For example, Black *et al.* [3] learned an offline subspace appearance model to represent object, however, the offline learned appearance model is hard to deal with the appearance changes. To deal with this problem, some online learning models have been proposed such as the WLS tracker [11] and IVT [2] tracker. Adam *et al* [12] utilized multiple instances to update an appearance model which is robust to partial occlusions. Those generative models require numerous samples to learn appearance feature, which will greatly increase the complexity. Furthermore, these models do not take background information [10] into account in which some useful information can help to visual tracking.

Discriminative models regard visual tracking as a classification task [13] in which a classifier is trained to separate targets from their surrounding background within a local region [20]. The *l1* tracker [10] was firstly proposed while many norm-related minimization problems need to be solved. Despite some advanced methods are proposed, it is still far away from being real-time. Boosting [6, 14] method has been introduced to object tracking in which weak classifiers with pixel-based features are combined. Collins *et.al* [13] demonstrated that discriminative features selection online can improve tracking performance. For example: Grabner *et al.* [6] proposed an online boosting feature selection method for object tracking. However, these above-mentioned discriminative algorithms [5-9] merely utilize one positive sample (the tracking result at the fore-frame) and multiple negative samples to update classifier. If the object location at previous frame is not precise, the positive sample will be noised and result in a sub optimal classifier update. Consequently, errors will be accumulated to cause tracking drift or failure [7]. In Ref [9], the MIL (multiple instance learning) model [7] is adopted to select features in a supervised learning model for object tracking, but, it has a great computational complexity. Recently, many improved tracking algorithms that based the MIL framework [7, 8, 9, 15, 16, 18] have been developed. For example: Zhang *et al.* proposed an ODFS tracker [16], which adopts a new strategy to select discriminative features and improved the performance to some extent. However, the classifier may inaccurate because it does not take the importance of positive samples into consideration during the feature selection strategy, moreover, this method only adopts the reverse gradient of sole *correct* positive sample to replace the average of whole positive samples during the objective function optimization, which may lead to less discriminative features to be selected.

In this paper, based on the ODFS tracker's framework, we proposed a weighted online discriminative feature selection tracker that integrates the sample's contribution into feature selection strategy. A new probability function integrating the weight of instances is present and then an efficient method is adopted to approximately optimize the objective function. Experimental results on challenging video sequences demonstrate the superior performance of our method in robustness and precision to some state-of-the-art tracking methods.

The paper is organized as follows: In Section 2, we firstly introduces the framework [7, 8, 9, 15, 16] of this proposed tracking algorithm and explains some related works, Section 3 gives the principle of our method and analysis its advantages over other methods in details. Section 4 presents the detailed experiment setup and demonstration of our tracking performance. Finally, a conclusion is given in Selection5.

2 Tracking by Detection and Related Works

2.1 System Overview

Let $l_t(\mathbf{x})\in R^2$ denotes the location of sample $\mathbf{x}$ at tth frame. The basic flow of tracking by detection is described as follow: based on the tracking result $l_{t\text{-}1}(\mathbf{x}_0)$ at t-1th frame, when tth frame is coming, the tracker firstly crops some candidate samples [25] from set $X^{\gamma}=\{x\,||l_t(\mathbf{x})-l_{t\text{-}1}(\mathbf{x}_0)|<\gamma\}$ with a relativity large radius γ surrounding the tracking result $l_{t\text{-}1}(\mathbf{x}_0)$; then the coped samples are classified and a sample (location $l_t(\mathbf{x}_0)$) with the maximum confidence is assume to be the new object at tthframe; finally, the classifier is updated by the positive and negative samples which cropped from region $X^{\alpha}=\{\mathbf{x}\,||l_t(\mathbf{x})-l_t(\mathbf{x}_0)|<\alpha\}$ and $X^{\xi,\beta}=\{\mathbf{x}\,|\,\xi<|l_t(\mathbf{x})-l_t(\mathbf{x}_0)|<\beta\}$ respectively. Based on the object location $l_t(\mathbf{x}^*)$ at tth frame, the tracking system is running by repeating the above-mentioned procedures.

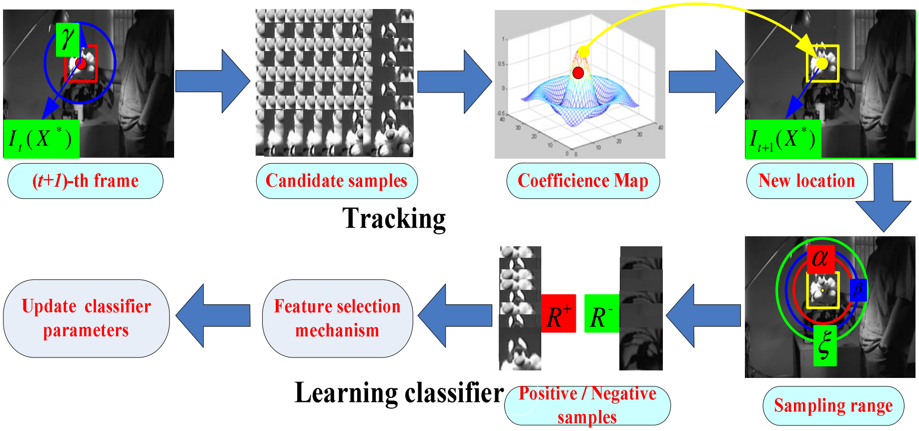


Fig. 1. The basic flow of tracking by detection algorithm

Algorithm 1. Tracking by detection

```
Input: $tth$ video frame
1. Get a set of candidate image samples: $X^{\gamma}=\{x\,|\,|l_t(\mathbf{x})-l_{t-1}(\mathbf{x}_0)|<\gamma\}$, where $l_{t-1}(\mathbf{x}_0)$ is the target location at $t$-1$th$ frame, and extract features $\{f_k(\mathbf{x})\}_{k=1}^{K}$ for each image samples.
2. Apply classifier in (2) to each candidate samples and find the sample location $l_t(\mathbf{x}_0)$ with the maximum confidence.
3. Get two sets of image samples $X^{\alpha}=\{\mathbf{x}\,|\,|l_t(\mathbf{x})-l_t(\mathbf{x}_0)|<\alpha\}$ and $X^{\xi,\beta}=\{\mathbf{x}\,|\,\xi<|l_t(\mathbf{x})-l_t(\mathbf{x}_0)|<\beta\}$ for positive samples and negative respectively.
4. Select features by the proposed feature selection strategy and update the classifier parameters according to (3) and (4).
Output: tracking location $l_t(\mathbf{x}_0)$ and classifier parameters.
```

2.2 Classification

In the tracking by detection algorithm, classifier [16] estimates the confidence of each sample via it's posterior probability function:

$$c(\mathbf{x})=\mathrm{p}(\mathrm{y}=1\,|\,\mathbf{x})=\sigma(\mathrm{h}_K(\mathbf{x})) \tag{1}$$

where $\mathbf{x}$ is a sample and $y\in\{0,1\}$ is a binary variable that represents the sample as positive or negative, $\sigma(z)=1/(1+e^{-z})$ is a sigmoid function and the classifier h_K is a liner combination of weak classifiers. Then the appearance model based on classifier $h_K(\mathrm{x})$ is defined as

$$h_K(\mathbf{x})=\log\left(\frac{\prod_{k=1}^{K}p(f_k(\mathrm{x})\,|\,\mathrm{y}=1)\mathrm{P}(\mathrm{y}=1)}{\prod_{k=1}^{K}p(f_k(\mathrm{x})\,|\,\mathrm{y}=0)\mathrm{P}(\mathrm{y}=0)}\right)=\sum_{k=1}^{K}\phi_k(\mathbf{x}) \tag{2}$$

where function $\phi_k(\mathbf{x})=\log\left(\frac{p(f_k(\mathrm{x})\,|\,\mathrm{y}=1)}{p(f_k(\mathrm{x})\,|\,\mathrm{y}=0)}\right)$ is a weak classifier, $f(\mathbf{x})=(f_1(\mathbf{x}),\ldots,f_K(\mathbf{x}))^T$ is a haar-like feature vector [7,16,17,18] for sample $\mathbf{x}$ and K is the number of features to be selected.

2.3 Classifier Construction and Update

The location distribution $p(f_k\,|\,\mathrm{y}=1)$ and $p(f_k\,|\,\mathrm{y}=0)$ in the classifier $h_K(\bullet)$ are assumed to be Gaussian distributed like the CT tracker [18] method with four parameters $(\mu_k^+,\sigma_k^+,\mu_k^-,\sigma_k^-)$ and they are defined as follows：

$$p(f_k\,|\,\mathrm{y}=1)\sim N(\mu_k^+,\sigma_k^+),\; p(f_k\,|\,\mathrm{y}=0)\sim N(\mu_k^-,\sigma_k^-) \tag{3}$$

The parameters $(\mu_k^+, \sigma_k^+, \mu_k^-, \sigma_k^-)$ in (3) are incrementally updated as follows

$$\begin{aligned} \mu_k^+ &\leftarrow \eta\mu_k^+ + (1-\eta)\mu^+ \\ \sigma_k^+ &\leftarrow \sqrt{\eta(\sigma_k^+)^2 + (1-\eta)(\sigma_k^+)^2 + \eta(1-\eta)(\mu_k^+ - \mu^+)^2} \end{aligned} \tag{4}$$

where $\sigma^+ = \sqrt{\frac{1}{N}\sum_{i=0|y=1}^{N-1}(f_k(\mathbf{x}_i) - u^+)^2}$, $\mu^+ = \frac{1}{N}\sum_{i=0|y=1}^{N-1} f_k(\mathbf{x}_i)$ and N is the number of positive samples. Similarly, the tracker updates the parameters (μ_k^-, σ_k^-). The above-mentioned (3) and (4) can be easily deduced by maximum likelihood function and η is a learning rate to adjust the effect between the previous frames and the current one.

A feature pool with M ($M>K$) features is maintained during learning procedure. As demonstrated in Ref [5], online selection of the discriminative features between object and background can improve the tracking performance significantly. Tracking task is to detect the sample that with the maximum confidence based on the selected features.

2.4 Related Works on Feature Selection Strategy

Recently, Zhang et al. [16] has proposed an online discriminative feature selection technique to improve the tracking performance to some extent. The ODFS tracker selects a subset of weak classifier to maximizes the average confidence of positive samples while suppressing the average confidence of negative samples. However, Zhang et al [16] made a rough simplification by representing the average gradient of all positive samples with the reverse gradient of classifier score of object location at previous frame, which may lead the ODFS tracker easily select less effective features. Moreover, the appearance model does not consider the different contributions of the positive samples into the feature selection procedure, which may cause drafting when the target location is not precise at previous frame. In the next section, we proposed an efficient online feature selection method which is a sequential forward selection method [17] where the number of feature combination is MK, thereby facilitating real-time processing.

3 Weighted Online Discriminative Feature Selection

3.1 Principle of Our Method

Similarly, the proposed feature selection strategy selects a subset of weak classifiers $\{\phi_k\}_{k=1}^K$ that have highest classification score between positive and negative samples from the feature pool Φ. These positive samples have different distance to the fore-tracking result and they also make different contributions to the objective function, so we assure that samples near the *correct* sample contribute more to the objective function than those far from it. Therefore, unlike the Noisy-OR model [16] adopted by

ODFS tracker, our method naturally integrates the sample importance into feature selection strategy and define the sample importance as follows

$$w_{i0} = {}^{1}\!/_{c}\, e^{-|l(\mathrm{x}_i)-l(\mathrm{x}_0)|} \tag{5}$$

where $l(\bullet)\in\mathrm{R}$ indicates the location and c is a normalization constant. Simply, negative samples are considered as making the same contribution to the objective function because all of the negative samples are far away from the center of *correct* sample. Finally, we define a margin as the difference of the total confidence of weighted positive samples minus the total confidence of negative samples. Then the objective function can be formed as follows

$$\begin{aligned} E_{margin} &= {}^{1}\!/_{N}\sum_{i=0}^{N-1} w_{i0}\sigma(\sum_{k=1}^{K}\phi_k(\mathbf{x}_i)) - {}^{1}\!/_{L}\sum_{i=N}^{N+L-1}\sigma(\sum_{k=1}^{K}\phi_k(\mathbf{x}_i)) \\ &\approx {}^{1}\!/_{N}(\sum_{i=0}^{N-1} w_{i0}\sigma(\sum_{k=1}^{K}\phi_k(\mathbf{x}_i)) - \sum_{i=N}^{N+L-1}\sigma(\sum_{k=1}^{K}\phi_k(\mathbf{x}_i))) \end{aligned} \tag{6}$$

where N and L is the number of positive samples and negative samples respectively, $\sigma(z)=1/(1+e^{-z})$ is a sigmoid function. Each selected feature must maximize the margin function, thus the weak classifier can be selected as follows

$$\phi_k = \arg\max_{\phi\in\Phi}(\sum_{i=0}^{N-1} w_{i0}\sigma(h_{k-1}(\mathbf{x}_i)+\phi(\mathbf{x}_i)) - \sum_{i=N}^{N+L-1}\sigma(h_{k-1}(\mathbf{x}_i)+\phi(\mathbf{x}_i))) \tag{7}$$

where h_{k-1} is a liner combination of previous *k-1* weak classifiers. We define $g_{k-1}(\mathbf{x})$ is the inverse gradient (the steepest descent direction) of the posterior probability function $\sigma(h_{k-1})$ with respect to classifier h_{k-1}. We introduced the gradient into objective function in a way that similar to the method in Ref [19], then the objective function can be translated into follows

$$\phi_k = \arg\max_{\phi\in\Phi}(\sum_{i=0}^{N-1} w_{i0}(g_{k-1}(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2 + \sum_{i=N}^{N+L-1}(-g_{k-1}(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2 \tag{8}$$

where the gradient function $g_{k-1}(\mathbf{x})$ is defined as

$$g_{k-1}(\mathbf{x}) = -\frac{\partial\sigma(h_{k-1}(\mathbf{x}))}{\sigma h_{k-1}} = -\sigma(h_{k-1}(\mathbf{x}))(1-\sigma(h_{k-1}(\mathbf{x}))) \tag{9}$$

However, the constraint between the selected ϕ_k and the inverse gradient direction g_{k-1} is very strong because ϕ_k is limited to the classifier pool Φ, which will bring huge computation. To alleviate these problems, we proposed a multiple grades strategy which divides the positive samples into three different grades. Within the radius of cropping positive samples, we set three different crop grades to get variety of positive samples. The radius difference between each grade is two pixels. Sample that near closest to the fore-tracking location center is adopt to replace the samples of corresponding grade for inverse gradient calculation. The weight of each grade can be calculated by the distance between the tracking location and sample that near closest to the tracking location in corresponding class. So the objective function can be formulated into

$$\phi_k = \underset{\phi\in\Phi}{\arg\max}\Big(\sum_{i=0}^{N_1-1} w_{i0}(g_{k-1}(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2 + \sum_{i=N_1}^{N_2-1} w_{i0}(g_{k-1}(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2 + \sum_{i=N_2}^{N-1} w_{i0}(g_{k-1}(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2 + L(\bar{\bar{g}}_{k-1}-\bar{\bar{\phi}})^2\Big) \tag{10}$$

where $\bar{\bar{g}}_{k-1}=1/L\sum_{i=N}^{N+L-1} g_{k-1}(\mathbf{x}_i)$ indicates the average inverse gradient of classification score of fore-combined *k-1* classifiers, $\bar{\bar{\phi}}=1/L\sum_{i=N}^{N+L-1}\phi(\mathbf{x}_i)$ indicates the average classifier score of all negative samples. Then we take a simplification strategy into the optimization: average the inverse gradient of fore *k-1* classifications by replacing $\sum_{i=0}^{N_1-1} w_{i0}g_{k-1}(\mathbf{x}_i)$, $\sum_{i=N_1}^{N_2-1} w_{N_10}g_{k-1}(\mathbf{x}_{N_1})$ and $\sum_{i=N_2}^{N-1} w_{N_20}g_{k-1}(\mathbf{x}_{N_2})$ with $w_{00}g_{k-1}(\mathbf{x}_{00})$, $w_{N_10}g_{k-1}(\mathbf{x}_{N_10})$, and $w_{N_20}g_{k-1}(\mathbf{x}_{N_20})$ respectively, then only the current classifier should be applied to all of the positive samples and the feature selection criterion becomes:

$$\phi_k = \underset{\phi\in\Phi}{\arg\max}\Big(E_{WODFS}(\phi) = w_{00}(N_1\,g_{k-1}(\mathbf{x}_0)-\sum_{i=0}^{N_1}\phi(\mathbf{x}_i))^2 + w_{N_10}((N_2-N_1)\,g_{k-1}(\mathbf{x}_{N_1})-\sum_{i=N_1-1}^{N_2}\phi(\mathbf{x}_i))^2 + w_{N_20}((N-N_2)\,g_{k-1}(\mathbf{x}_{N_2})-\sum_{i=N_2-1}^{N}\phi(\mathbf{x}_i))^2 + L(-\bar{\bar{g}}_{k-1}-\bar{\bar{\phi}})^2\Big) \tag{11}$$

where $\mathbf{x}_{00},\mathbf{x}_{N_10},\mathbf{x}_{N_20}$ is the representative sample of different classes, $\mathbf{x}_{00}$ is the *correct* sample indeed , $\mathbf{x}_{N_10},\mathbf{x}_{N_20}$ are the samples that near closest to the tracking location in grade 2 and grade 3 respectively.

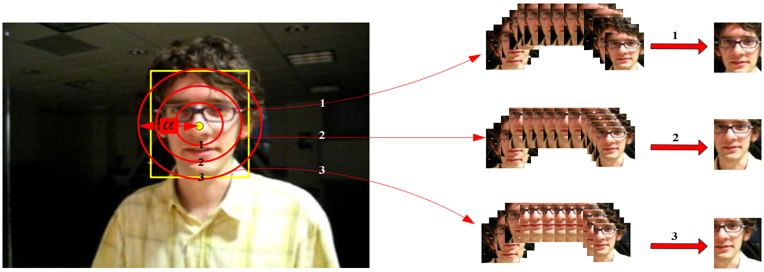


Fig. 2. The multiple grades strategy of positive samples

It is worth noting that the classifier must be applied to all weighted positive samples and negative samples when selecting current feature and the hierarchical strategy only used for the inverse gradient of the former *k-1* features. Moreover, the average strategy of each grade samples is adopted to reduce computation. In addition, the weighted gradient of the most *correct* sample in different grades helps to select effective features which can reduce sample ambiguity errors. When a new frame arrives, we update all the weak classifiers in the pool Φ in parallel, and select *K* weak classifiers sequentially based on the strategy in (11). The main steps of the proposed feature selection algorithm are summarized in Algorithm 2.

Algorithm2. Advanced Online Discriminative Feature Selection

```
Input: Samples $\{\mathbf{x}_i, y_i\}_{i=0}^{N+L-1}$ where $y_i \in \{0,1\}$
    Update the weak classifier pool $\Phi=\{\phi_m\}_{m=1}^{M}$ with samples $\{\mathbf{x}_i, y_i\}_{i=0}^{N+L-1}$.
    Update the weighted weak classifier outputs $\sum_{i=0}^{N-1}\phi(\mathbf{x}_i)$ and $\bar{\bar{\phi}}_m$ , $m$=1,...,$M$.
    Update weight for each positive samples by (5)
    Initialize $h_0(\mathbf{x}_i)=0$
    for $k$=1 to $K$ do
    Update $g_{k-1}(\mathbf{x}_i)$
        for $m$ =1 to $M$ do.
            $\phi_k = \arg\max_{\phi\in\Phi}(E_{WODFS}(\phi) = w_{00}(N_1\, \mathrm{g}_{k-1}(\mathbf{x}_0) - \sum_{i=0}^{N_1}\phi(\mathbf{x}_i))^2 + w_{N_1 0}((N_2 - N_1)\,\mathrm{g}_{k-1}(\mathbf{x}_{N_1}) - \sum_{i=N_1}^{N_2}$
                $+ w_{N_2 0}((N - N_2)\,\mathrm{g}_{k-1}(\mathbf{x}_{N_2}) - \sum_{i=N_2-1}^{N}\phi(\mathbf{x}_i))^2 + L(-\bar{\bar{g}}_{k-1} - \bar{\bar{\phi}})^2)$
        end for
    $m^* = \arg\max_m(\mathrm{E}_m)$ .
    $\phi_k \leftarrow \phi_{m^*}$ .
    $h_k(\mathbf{x}_i) \leftarrow \sum_{j=1}^{k}\phi_j(\mathbf{x}_i)$ , $h_k(\mathbf{x}_i) \leftarrow h_k(\mathbf{x}_i) / \sum_{j=1}^{k}|\phi_j(\mathbf{x}_i)|$
    end for
Output: Strong classifier $h_K(\mathbf{x}) = \sum_{k=1}^{K}\phi_k(\mathbf{x})$ and confidence function $P(\mathrm{y}=1|\mathbf{x}) = \sigma(h_K(\mathbf{x}))$.
```

3.2 Discussion

In this selection, we discuss the advantages of our method over other methods.

A . *Equal and Different Weight.* In (11), we give the sample that near the tracking location at current frame a larger weight based on the assumption that the tracking location at current frame is the most *correct* positive sample. This assumption is adopted in most generative models[2,3,4] and some discriminative models[5,6,7,8,9]. In fact, it is impossible to ensure a complete drift free tracker without any prior models and learning classifier online. However, the proposed tracker can deal with the drift problem based on a weighted feature selection strategy which maximizes the total classification confidence of weighted positive samples while suppressing the total classify confidence of negative ones. If different positive samples are given the same weight, the classifier can become confused that it cannot select discriminative features because each positive samples contributes equally to the objective function.

B . *Sample Ambiguity Problem.* Babenko.*et al.* [7] recently demonstrated that the location ambiguity problem can be alleviated with online multiple instance learning, but MIL tracking is still not stable in some challenging tracking tasks. There may be several factors. Firstly, the Noisy-OR model [16] adopted in ODFS tracker could not eliminate error that brought in by uncertainty samples, and may select less effective features; secondly, the classifier is only trained by the binary labels without considering the different contributions of these samples. On the contrary, the feature selection criterion in our method explicitly relates the classifier score with the importance of samples. Therefore, the ambiguity problem can be better deal with.

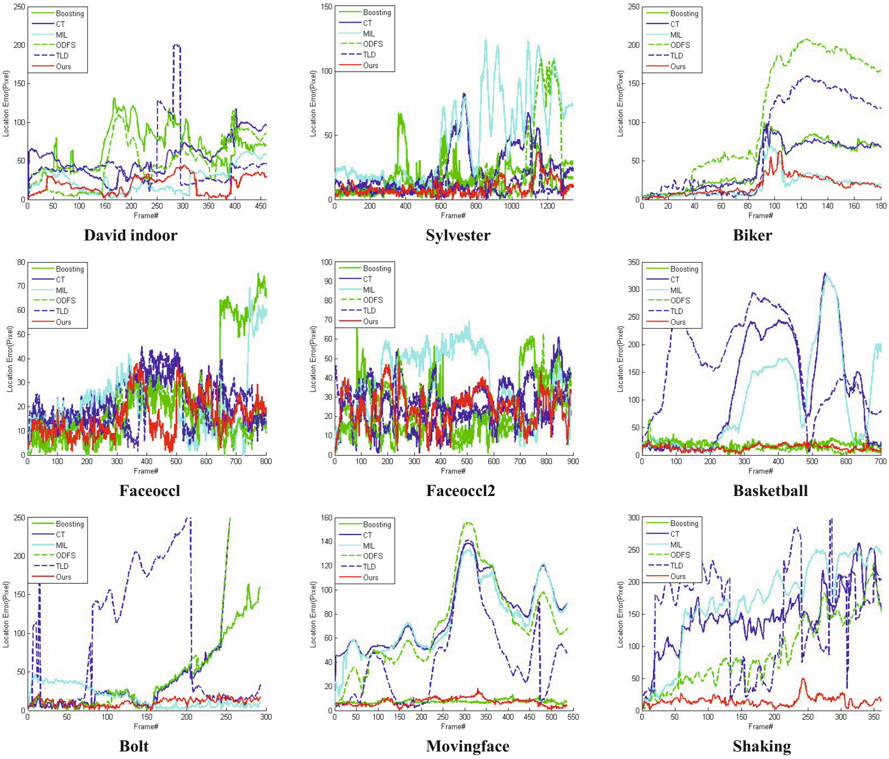
C *. Advantages of Our Method over ODFS Tracker.* The ODFS tracker adopts a rough simplification that only using the sole *correct* sample to represent the average of whole positive samples while some noise may included when drafting. Our method divides the positive samples into three classes and weights the contribution of positive samples to reduce error. Thus the proposed method can select more effective feature than ODFS tracker, especially in case of drastic illumination variation and pose changes.

4 Experimental Results

In this section, we use a radius (6 pixels) to crop positive samples. A small α can generates incorrect samples when drafting while a large α can make positive samples much more variety which are sufficient to avoid noise. The inner and outer radius for the set $X^{\xi,\beta}$ that generates negative samples are set as $\xi=12$ $\beta=48$ respectively. Then we randomly select a set of 50 negative samples from the set $X^{\xi,\beta}$. The radius for searching new target location in the next frame is set as 25 which can fully include all candidate targets because the target motion between two consecutive frames is often smooth. We set K ,M and c as 15,50 and 4 respectively. A small learning rate can make the tracker quickly adopts to the fast appearance changing while a large learning rate can reduce the likelihood that the tracker drifts off the target. The best learning rate can be set as $\eta=0.80$ in experiments. For other competing algorithms, we use the original source codes or binary codes released by the authors. Our tracker is implemented in MATLAB and runs at 25 fps on Intel Dual-Core 1.7GHz CPU with 2.0 GB RAM. The videos used in the experiments can be found at http://youtube/3UobcBa-V1Q.

4.1 Quantitative Evaluation

To evaluate the performance of the algorithm, two performance indexes--*center location error* and *success rate* [18] are adopted to evaluate the proposed method with other 5 trackers [7,15,16,18,23]. The *center location error* is measured as the Euclidean distance between the center location of the tracked target and ground truth. *Success rate* indicates the percentage of successful frames whose overlap score is larger than the half of $|r_g|$. The overlap score is defined as $OS=\frac{|r_t \cap r_g|}{|r_t \cup r_g|}$, where r_t indicates the tracked bounding box, r_g represents the ground truth bounding box, $\cap$ and $\cup$ represent the intersection and union of two regions, and $|\bullet|$ denotes the number of pixels in the region. We draw *center location error* plots and present average *center location error* and s*uccess rate* in tabular form to show super performance over other methods. Overall, our method favorably performs against other trackers.



David indoor

Sylvester

Biker

Faceoccl

Faceoccl2

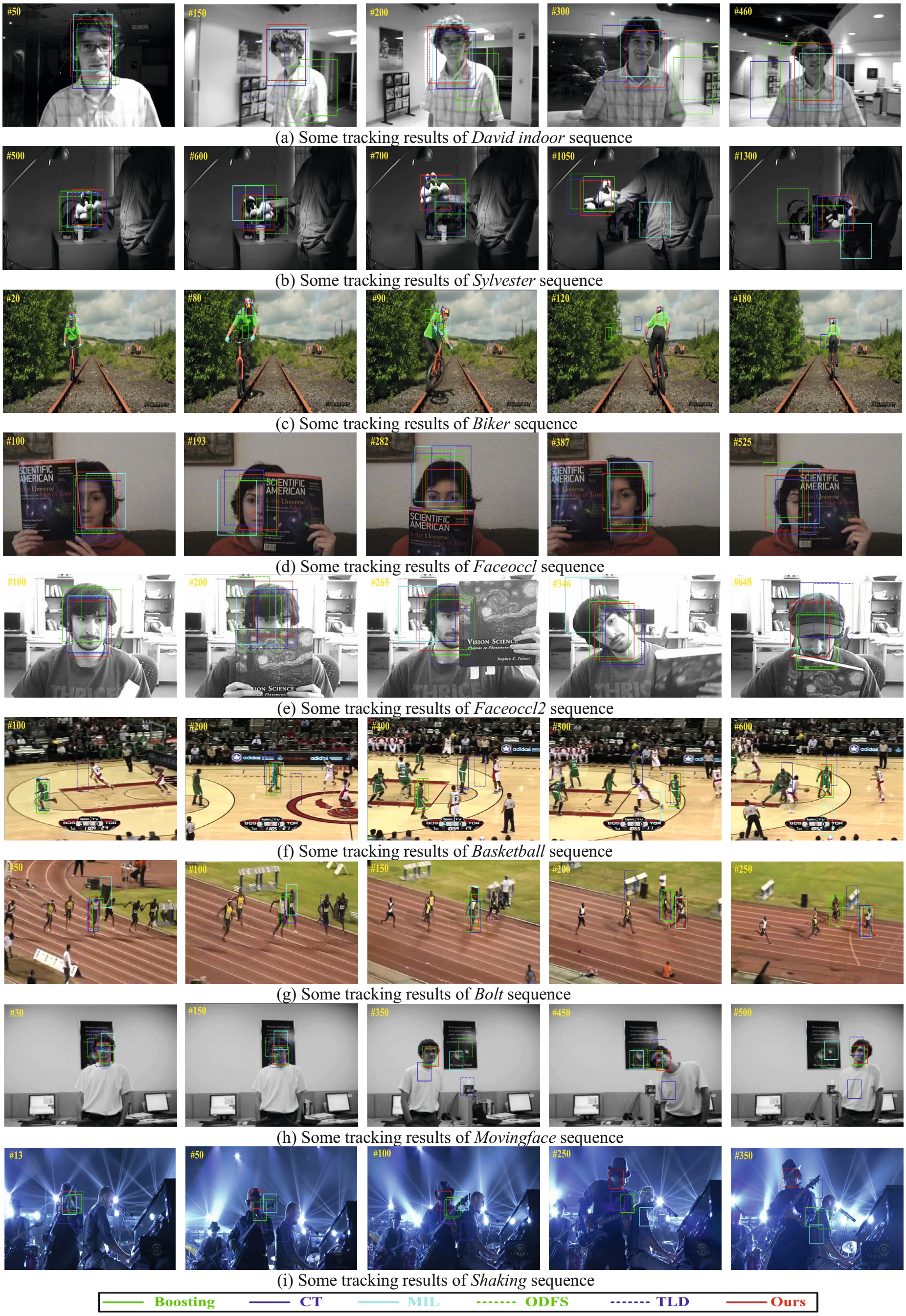
Basketball

Bolt

Movingface

Shaking

Fig. 3. Center Location Error of 9 challenging sequences


(a) Some tracking results of *David indoor* sequence

(b) Some tracking results of *Sylvester* sequence

(c) Some tracking results of *Biker* sequence

(d) Some tracking results of *Faceoccl* sequence

(e) Some tracking results of *Faceoccl2* sequence

(f) Some tracking results of *Basketball* sequence

(g) Some tracking results of *Bolt* sequence

(h) Some tracking results of *Movingface* sequence

(i) Some tracking results of *Shaking* sequence

Fig. 4. Some tracking results in 9 challenging sequences

Table 1. Average Center Location Error (ACLE)

Video	**Boosting**	**CT**	**MIL**	**ODFS**	**TLD**	**Ours**
David indoor	67	54	25	47	44	**19**
Sylvester	16	22	46	22	9	**8**
Biker	45	44	18	109	72	**17**
Faceoccl	23	22	20	**15**	20	15
Faceoccl2	24	24	37	**22**	26	24
Basketball	18	115	103	14	165	**12**
Bolt	42	73	17	74	89	**10**
Moving face	**7**	80	76	72	43	**7**
Shaking	106	145	169	91	147	**14**
Average ACLE	38	64	56	51	68	**14**
Average fps	8	**33**	10	30	9	25

The **Bold** fonts indicate the best performance in this test.

Table 2. Average Success Rate (ASR) (%)

Video	**Boosting**	**CT**	**MIL**	**ODFS**	**TLD**	**Ours**
David indoor	32	64	71	68	47	**81**
Sylvester	80	83	53	94	81	**99**
Biker	66	74	**75**	35	30	73
Faceoccl	82	76	**94**	**94**	87	90
Faceoccl2	80	77	82	**90**	82	90
Basketball	71	35	4	69	0	**87**
Bolt	15	45	90	48	14	**92**
Moving face	**78**	47	27	24	25	78
Shaking	0	25	1	73	0	**74**
Average ASR	56	58	56	66	40	**84**

The **Bold** fonts indicate the best performance in this test.

4.2 Qualitative Evaluation

Scale and Pose Change: Although our tracker only estimates the transnational motion similar to most state-of-art algorithms (Boosting, MIL, CT), it can also handle scale and orientation changes because of the compressed Haar-like features [18]. In the *David indoor* sequence, the target has big scale and pose changes and illumination variation, it is noting that the MIL,CT and ODFS trackers perform well in some extent on this sequence while the Boosting, TLD tracker drift. The compressed feature enable MIL and our tracker to handle the scale and pose changes well, and our tracker yields more accurate results (frame#150, #200, #300) than the ODFS tracker because it can select more informative features to separate target from background by eliminating errors.

The CT tracker suffers some drifts because it does not select features online. The TLD tracker fails to track the object mainly because it relies heavily on the visual information in the first frame to re-detect the object. Moreover our method performs well on the *Sylvester* and *Biker* sequence in which the targets undergo significant pose changes.

Background Clutter and Pose Variation: For the sequence (*Bolt, Basketball*) shown in Fig4(f,g), the pose of the object change gradually and background are full of clutter. Only MIL and our tracker perform well on the video *Bolt* (frame#150, #200, #250), to deal with cluttered background, the MIL tracker set some instances from positive and negative instance bags respectively to learn the classifier to resist background interference; the boosting tracker is a generative model that does not take the background information into consideration and it drifts to the background. The features maintained in CT and ODFS may be contaminated by clutter background, which will result in the tracking failure. In *Basketball,* the Boosting, ODFS and our tracker perform well because they select discriminative feature for object representation which can well handle pose variation and shape deformation. The MIL, CT and TLD trackers do not perform well as generative models are less effective to account for appearance change caused by large shape and pose variation, thereby making the method drift away to similar objects.

Occlusion and Rotation: The target object in sequences *Faceoccl* and *Faceoccl2* undergoes large pose variation and heavy occlusion. In test video *Faceoccl*, the ODFS and our tracker perform well (frame#193, #282, #387) due to their efficient online feature selection strategy. The CT and MIL tracker extract some positive and negative samples to update classifier while they do not take informative features into consideration. In *Faceoccl2*, the ODFS and our tracker can handle rotation well (frame#200, #346, #648) because the tracker can also extract informative features to update classifier when target rotated while the other trackers drift seriously on this sequence.

Large Illumination Change and Pose Variation: For the *shaking* sequence shown in Fig4(i), the illumination and pose of the object both change gradually. The appearance of the singer's head in the *shaking* sequence changes significantly due to large variation of illumination and head pose. The CT and TLD tracker fails (frame#15) to track the head when the stage light drastically changes, whereas our tracker can accurately locate the target. The Boosting and ODFS tracker drift when the heavy illumination change and pose variation as shown in frame#50, #100, MIL tracker fails to track the object when heavy illumination change at frame #50 while our tracker is able to adjust the classifier quickly to appearance change and thus the proposed method performs well when illumination change and pose variation.

5 Conclusion

In this paper, a robust tracker based on an online updating appearance is proposed, which naturally integrate the positive sample importance into learning procedure.

The proposed method assumes the object location at current frame is the *correct* sample with which to make each sample contribute differently to the learning strategy: the closer the sample is to the center of *correct* sample, the more it contributes to the objective function. Experiments demonstrate that the classifier learned by the approach adopted in this paper is much more stable and robust than those in ODFS algorithm. The proposed method performs well on challenging sequences indicate the superiority over other state-of-the art algorithms in terms of accuracy and robust.

Acknowledgments. This research work was supported by the National 863 Program (No. 2012AA112312), National Nature Science Foundation of China (No.61175075) and Doctoral Program Foundation of Institutions of Higher Education of China, (No.20110161120006).

References

[1] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Survey 38 (2006)

[2] Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77(1), 125–141 (2008), 1, 2, 7, 8

[3] Jepson, A., Fleet, D., EI-Maraghi, T.: Robust online appearance model for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1296–1311 (2003)

[4] Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: International Conference on Computer Vision, pp. 1436–1443 (2009)

[5] Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1631–1643 (2005)

[6] Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In: British Machine Vision Conference, pp. 47–56 (2006)

[7] Babenko, B., Yang, M., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Transaction on Pattern Analysis and Machine Intelligence 33, 1619–1632 (2011)

[8] Li, H., Shen, C., Shi, Q.: Real-time visual tracking using compressive sensing. In: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1305–1312 (2011)

[9] Zhang, K., Song, H.: Real-time visual tracking via online weighted multiple instance learning. Pattern Recognition 46(1), 397–411 (2013)

[10] Zhong, W., Lu, H., Yang, M.: Robust object tracking via sparsity based collaborative model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1838–1845 (2012)

[11] Jepson, A., Fleet, D., EI-Maraghi, T.: Robust online appearance models for visual tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence 25, 1296–1311 (2003)

[12] Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 798–805 (2006)

[13] Avidan, S.: Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1064–1072 (2004)

[14] Avidan, S.: Ensemble tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence 29, 261–271 (2007)

[15] 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)

[16] Zhang, K., Zhang, L.: Real-tine object tracking via online discriminative feature selection. IEEE Transaction on Image Processing, 4664–4677 (2013)

[17] Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature mining for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)

[18] Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)

[19] Friedman, J.: Greedy function approximation: A gradient boosting machine. The Annas of Statistics 29, 1189–1232

[20] Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: A benchmark. In: CVPR (2013)

[21] Salti, S., Cavallaro, A., Di Stefano, L.: Adaptive appearance modeling for video tracking: Survey and evaluation. IEEE Transaction on Image Processing 21(10), 4311–4348 (2012)

[22] Wang, S., Lu, H., Yang, F., Yang, M.: Superpixel tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1323–1330 (2011)

[23] Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 49–56 (2010)

[24] Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via spare representation. IEEE Transaction on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)

[25] Liu, L., Fieguth, P.: Texture classification from random features. IEEE Transaction on Pattern Analysis and Machine Intelligence 34(3), 574–586 (2012)