Treat samples differently: Object tracking with semi-supervised online CovBoost

Guorong Li[1] Lei Qin[2] Qingming Huang[1,2] Junbiao Pang[2] Shuqiang Jiang[2]
[1]Graduate University of Chinese Academy of Sciences(CAS),Beijing 100049, China
[2]Key Lab of Intell. Info. Process., Inst. of Computi. Tech., CAS, Beijing 100190, China
{grli,lqin,qmhuang,jbpang,sqjiang}@jdl.ac.cn

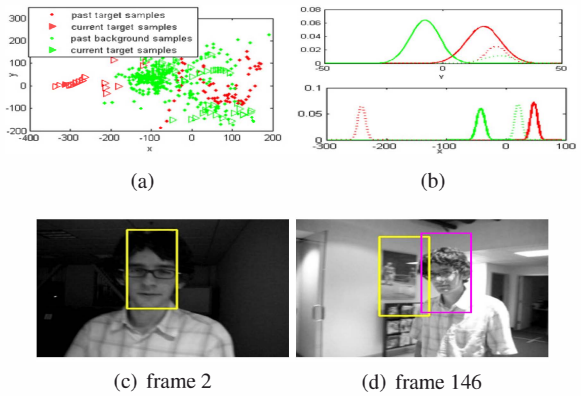
Abstract

Most feature selection methods for object tracking assume that the labeled samples obtained in the next frames follow the similar distribution with the samples in the previous frame. However, this assumption is not true in some scenarios. As a result, the selected features are not suitable for tracking and the "drift" problem happens. In this paper, we consider data's distribution in tracking from a new perspective. We classify the samples into three categories: auxiliary samples (samples in the previous frames), target samples (collected in the current frame) and unlabeled samples (obtained in the next frame). To make the best use of them for tracking, we propose a novel semi-supervised transfer learning approach. Specifically, we assume only target samples follow the same distribution as the unlabeled samples and develop a novel semi-supervised CovBoost method. It could utilize auxiliary samples and unlabeled samples effectively when training the best strong classifier for tracking. Furthermore, we develop a new online updating algorithm for semi-supervised CovBoost, making our tracker handle with significant variations of the tracked target and background successfully. We demonstrate the excellent performance of the proposed tracker on several challenging test videos.

1. Introduction

Object tracking has been a hot research topic for decades and researchers have made great progresses (e.g. [5, 11, 3]) during the past years. However, there still exist many open problems facing complicated variations in target's appearance and motion, as well as background. In [4, 1], object tracking is considered as a local discrimination problem: foreground and background. Good features can be used to discriminate target from background easily and accurately. Thus feature selection is a challenging, but very important problem for object tracking.

To be adaptive to dynamic background and object's variations, many online feature selection methods such as [4, 8, 22] are proposed. In [4], variance ratio is used to evaluate feature's discriminative ability and the best k features are selected to describe the target's appearance. Meanwhile due to its good performance and computation efficiency, online boosting methods ([8, 22]) become one of the most popular feature selection algorithms for object tracking. For example, in [8], based on the obtained labeled samples, a classifier is trained and updated through online boosting. To alleviate drift problem and make use of prior, semi-supervised methods, such as [20, 24, 9, 15] are developed for tracking. Recently, multiple instance learning methods such as [2, 25] are developed to resolve the ambiguities where to take positive updates during tracking. Different from other semi-supervised methods, in [12], a very interesting paradigm called "P-N learning", which is guided



(a) (b)

(c) frame 2 (d) frame 146

Figure 1. A video example in which appearances of the tracking target and background varies significantly because of lightness and motion. We extract 100 Haar features for every sample and then reduce the dimension to 2 using PCA. The red and green dots denote target samples and background samples collected from frame 3 to frame 142. Red and green triangles denote samples collected in 10 frames around frame 146. Apparently, dot samples and triangle samples follow the different distributions as shown in (b). In (c), tracking target is labeled with the yellow rectangle. So training with dot samples, boosting method [8] generate wrong classification result on triangle samples. (d) shows its tracking result (in yellow) and ours (in magenta).

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

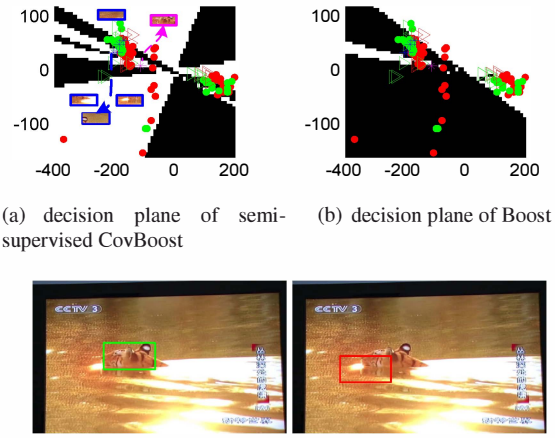
by positive and negative constraints, is proposed to restrict the labeling of unlabeled data. It could guarantee the improvement of the classifier during learning process. However, we would like to point out that these approaches make a common assumption: in tracking, the samples they used for training follow the same distribution with the samples they aim to classify. However, when target's appearance or background varies greatly or continuously, the underlying data distribution keeps changing too and is not necessarily from i.i.d. [7]. In other words, there is difference between previous and current samples. Perhaps, in human eyes, the difference is not very large, but in this case features selected using existing methods, such as [8, 22] may generate wrong results on the target samples, although they could lead to the minima classification error on the past samples. So tracking results become inaccurate and finally "drift" [17] happens. Fig. 1 shows a very good example.

In object detection, some works such as [18, 23] have developed efficient transfer learning methods [19] to deal with the case that the training and testing data follow different distributions. However, this case has not been studied for object tracking and it does exist in many situations such as videos containing fast moving targets or motion blur, low frame-rate videos and home videos self-recorded with hand held camera. So it is necessary to investigate this case for tracking. In this paper, we consider data's distribution form a new angle and introduce "Covariate Shift" problem into tracking. Then we propose a new semi-supervised online boosting algorithm for solving this problem. We classify the samples in object tracking into three categories: labeled auxiliary samples, labeled target samples and unlabeled target samples as shown in Fig. 3. We assume that the labeled auxiliary samples and target samples are under covariate shift. To select the best features for discriminating the unlabeled target samples and obtain tracking results, we develop a semi-supervised online boosting method. Experimental results demonstrate that it can alleviate "drift" to some extent and thus could achieve superior performance. See Fig. 2 for a concrete example.

The rest of the paper is organized as follows. In section 2, we provide a brief introduction to CovBoost and propose a new method for solving it. Then section 3 details the derivations of the proposed semi-supervised CovBoost considering into unlabeled target samples. After that, the online learning algorithm for tracking is elaborated in section 4. Experimental results are reported in section 5. Finally, we conclude the paper in section 6.

2. Covariate shift and CovBoost algorithm

Given labeled auxiliary set $\Im = \{< z_1, \ell_1 >, < z_2, \ell_2 >, \cdots, < z_{n_a}, \ell_{n_a} >\}$ and target training set $\chi^L = \{< x_1, y_1 >, < x_2, y_2 >, \cdots, < x_n, y_n >\}$. Let $D = \{z_1, z_2, \cdots, z_{n_a}, x_1, x_2, \cdots, x_n\}$. If $p(\ell|Z = z) = p(Y|X = z)$ for all $z \in D$, but $p(Z) \neq p(X)$, the difference between the two data sets is referred as "Covariate Shift" [19].



(a) decision plane of semi-supervised CovBoost (b) decision plane of Boost

(c) tracking result corresponding to (a) (d) tracking result corresponding to (b)

Figure 2. Best view in color. A toy problem: the dots, triangles and stars denote auxiliary samples, target samples and unlabeled samples, which are collected in the previous frames, current frame and next frame respectively. The red ones are positive while the green ones are negative. The pink star represents the image patch in (a) with pink rectangle corresponding to foreground in the next frame. The blue stars denote image patches in (a) with blue rectangle and they correspond to background in the next frame. We can see that, our proposed algorithm generate effective decision plane, which classifies the pink star into the positive class and generate accurate tracking result as shown in (c).

In [18], the difference between samples collected from different views is assumed to be covariate shift and CovBoost is proposed to solve view-adaptiveness in pedestrian detection. Based on the boosting method, CovBoost forms a strong classifier $H(\cdot)$ with weak classifiers $h_k(\cdot)$ by minimizing the extended loss function $L(H)$.

$$L_c(H, \Im, \chi^L) = \Sigma_{i=1}^{n} exp\{-2y_i H(x_i)\} + \sum_{j=1}^{n_a} \gamma(z_j, \ell_j) exp\{-2\ell_j H(z_j)\} \quad (1)$$

where $\gamma(z_j, \ell_j) = \frac{p(z_j, \ell_j | < z_j, \ell_j > \in \chi^L)}{p(z_j, \ell_j | < z_j, \ell_j > \in \Im)}$. In [18], $\gamma(x, y)$ is proved to be equivalent with $\frac{p(< x, y > \in \chi^L | x, y)}{p(< x, y > \in \Im | x, y)} \frac{p(< x, y > \in \Im)}{p(< x, y > \in \chi^L)}$.

Different from solving method in [18], using boosting method, we learn two classifiers $H_t(x)$ and $H_a(x)$ on χ^L and $\Im$ respectively . Then conditional distributions $p(< x, y > \in \chi^L | x, y)$ and $p(< x, y > \in \Im | x, y)$ could be modeled as $\frac{1}{1+exp\{-yH_t(x)\}}$ and $\frac{1}{1+exp\{-yH_a(x)\}}$ respectively. Following the derivation of AdaBoost [6], we first select $h_m(\cdot)$ and then solve α_m in a greedy method:

$$h_m(\cdot) = argmin_{h(\cdot)}\{\sum_{i=1}^{n} w_i^m \delta(h_m(x_i), y_i) + \Sigma_{j=1}^{n_s} \varpi_j^m \delta(h_m(z_j), \ell_i)\} \tag{2}$$

$$\alpha_m = \frac{1}{4}\ln\frac{1-\epsilon_m}{\epsilon_m} \tag{3}$$

where

$$w_i^m = exp\{-2y_i H_{m-1}(x_i)\} \tag{4}$$

$$\varpi_j^m = \gamma(z_j, \ell_j) exp\{-2\ell_j H_{m-1}(z_j)\} \tag{5}$$

$$\epsilon_m = \frac{\Sigma_{h(x_i)\neq y_i} w_i^m + \Sigma_{h(z_j)\neq \ell_j} \varpi_j^m}{\Sigma_{i=1}^{n} w_i^m + \Sigma_{j=1}^{n_a} \varpi_j^m}. \tag{6}$$

3. Semi-supervised CovBoost for Feature Selection

Generally, there are many unlabeled samples waiting to be classified. Although they don't have labels, they still could provide very useful information. Usually, samples with similar observations should share the same label. This is referred as data consistency in many works such as [26] and it has been proved to be helpful for tracking in [9, 25].

Let $S(\cdot,\cdot)$ denote the similarity function of two samples and similar to [16], the loss function is defined as:

$$L(H, \Im, \chi^L, U) = L_c(H, \Im, \chi^L) + L_u(H, \chi^L, U) \tag{7}$$

$$\begin{aligned} L_u(H, \chi^L, U) = & \frac{\sum_{x_i,u_j} S(x_i,u_j) exp\{-2y_i H(u_j)\}}{n n_u} \\ & + \frac{\sum_{z_i,u_j} \gamma(z_i,\ell_i) S(z_i,u_j) exp\{-2\ell_i H(u_j)\}}{n_a n_u} \\ & + \frac{\sum_{u_i,u_j} S(u_i,u_j) D(H,u_i,u_j)\}}{n_u^2} \end{aligned} \tag{8}$$

where $U = \{u_1, u_2, \cdots, u_{n_u}\}$ denotes unlabeled set, $D(H,u_i,u_j) = \frac{exp\{H(u_i)-H(u_j)\}+exp\{H(u_j)-H(u_i)\}}{2}$. $L_u(H,\chi^L,U)$ measuring the data inconsistency between labeled data and unlabeled data, and that between unlabeled samples. Simulating the underlying of AdaBoost, we first minimize the loss $L(\cdot)$ only with respect to $h_m(\cdot)$. After $h_m(\cdot)$ is determined, we then optimize α_m. Through first order Taylor expansion of $L(\cdot)$ at $H_{m-1}(\cdot)$, $h_m(\cdot)$ can be selected as following:

$$\begin{aligned} h_{s^m}(\cdot) = & argmin_{h(\cdot)}[\Sigma_{i=1}^{n} w_i^m \delta(h_m(x_i), y_i) \\ & + \Sigma_{j=1}^{n_s} \varpi_j^m \delta(h_m(z_j), \ell_j) \\ & - \frac{1}{n_u}\Sigma_{j=1}^{n_u}(p_m(u_j) - q_m(u_j)) h_m(u_j)] \end{aligned} \tag{9}$$

Algorithm 1: Semi-supervised Online CovBoost

Input: $H_t(\cdot)$, $H_a(\cdot)$ and training examples: $\{< x_1, y_1 >, \cdots, < x_N, y_N >\}$.

Output: Strong classifiers $H(\cdot)$

```
s^w_{m,k} = 0, s^c_{m,k} = 0, λ̃_m = 1, m = 1,2,···,M; k = 1,2,···,N_w;
for i = 1; i ≤ N; i++ do
    for m = 1; m ≤ M; m++ do
        if x is target sample then
            y_m = y_i, λ_m = λ̃_m;
        else
            if x is auxiliary sample then
                y_m = y_i, λ_m = r_{a,t}(x_i, y_m) λ̃_m;
            else
                y_m = sign(p̃_m(x_i) − q̃_m(x_i)),
                λ_m = |p̃_m(x_i) − q̃_m(x_i)|;
            end
        end
        for k = 1; k < N_w; k++ do
            if y_m ≠ h_{m,k}(x_i) then
                s^w_{m,k} += λ_m;
            else
                s^c_{m,k} += λ_m;
            end
            ε_{m,k} = s^c_{m,k} / (s^c_{m,k} + s^w_{m,k});
        end
        m^sel = arg_k min ε_{m,k}, α_m = ln (1−ε_{m,m^sel}) / ε_{m,m^sel};
        λ̃_m = λ̃_m exp{−2 y_m α_m h_m(x_i)};
    end
    α_m = α_m / Σ_{m=1}^{M} α_m;
end
return H(x) = Σ_{m=1}^{M} α_m h_{m^sel}(x));
```

$$\alpha_m = \frac{1}{4}\ln\left\{\frac{\begin{pmatrix}\Sigma_{h(u_j)=1} p_m(u_j) + \Sigma_{h(x_i)=y_i} w_i^m \\ +\Sigma_{h(u_j)=-1} q_m(u_j) + \Sigma_{h(z_j)=\ell_j} \varpi_j^m\end{pmatrix}}{\begin{pmatrix}\Sigma_{h(u_j)=-1} p_m(u_j) + \Sigma_{h(x_i)\neq y_i} w_i^m \\ +\Sigma_{h(u_j)=1} q_m(u_j) + \Sigma_{h(z_j)\neq \ell_j} \varpi_j^m\end{pmatrix}}\right\} \tag{10}$$

where

$$\begin{aligned} p_m(u_j) = & \frac{1}{n} exp\{-2H_{m-1}(u_j)\} \sum_{y_i=1} S(x_i,u_j) \\ & + \frac{1}{n_a} exp\{-2H_{m-1}(u_j)\} \sum_{\ell_i=1} \gamma(z_i,\ell_i) S(z_i,u_j) \\ & + \frac{1}{n_u} \sum_{i=1}^{n_u} S(u_i,u_j) exp\{H(u_i) - H(u_j)\} \end{aligned}$$

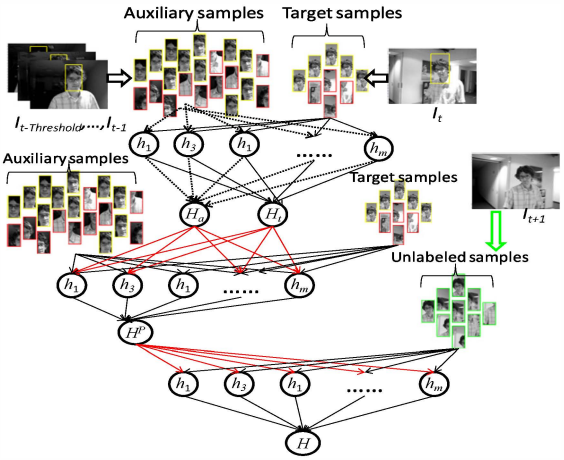


Figure 3. The framework of our semi-supervised online CovBoost tracker. The samples collected in the previous, current and next frames are all considered when training the strong classifier.

and

$$q_m(u_j) = \frac{1}{n} exp\{2H_{m-1}(u_j)\} \sum_{y_i=-1} S(x_i, u_j) + \frac{1}{n_a} exp\{2H_{m-1}(u_j)\} \sum_{\ell_i=-1} \gamma(z_i, \ell_i) S(z_i, u_j) + \frac{1}{n_u} \sum_{i=1}^{n_u} S(u_i, u_j) exp\{H(u_j) - H(u_i)\}.$$

Similar to [9], we consider $|\Im| \to \infty$ and boost the similarity $S(\cdot, \cdot)$ according to [10, 14],

$$S(x_i, u_j) = \frac{exp\{y_i H^p(u_j)\}}{exp\{H^p(u_j)\} + exp\{-H^p(u_j)\}} \tag{11}$$

$$\gamma(z_i, \ell_i) S(z_i, u_j) = \frac{exp\{\ell_i H^p(u_j)\}}{exp\{H^p(u_j)\} + exp\{-H^p(u_j)\}} \tag{12}$$

where $H^p(\cdot)$ is a prior classifier trained on $\Im$ and χ^L using CovBoost. Then $p_m(u_j)$ and $q_m(u_j)$ could be approximated as:

$$\tilde{p}_m(u_j) = \frac{exp\{-H_{m-1}(u_j)\} exp\{H^p(u_j)\}}{exp\{H^p(u_j)\} + exp\{-H^p(u_j)\}} \tag{13}$$

$$\tilde{q}_m(u_j) = \frac{exp\{H_{m-1}(u_j)\} exp\{-H^p(u_j)\}}{exp\{H^p(u_j)\} + exp\{-H^p(u_j)\}}. \tag{14}$$

We derive the online updating method for semi-supervised CovBoost and the details are displayed in Alg. 1.

4. Online Learning for Object Tracking

Fig. 3 shows the framework of our tracking method and Alg. 2 provides the detail procedure of the proposed tracking method. There are two crucial differences between our method and online boosting. One is that, we assume target samples follow the different distribution with auxiliary samples while the other boosting trackers treat them equally. The other is that we make use of unlabeled samples in the next frame.

Algorithm 2: On-line Semi-supervised CovBoost for Object Tracking

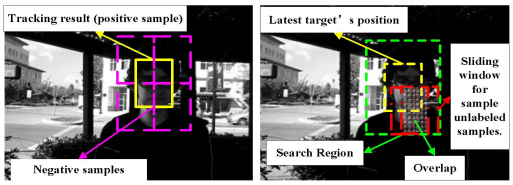
Input: video frame sequences $I_0, I_1, \cdots, I_F$
Output: tracking results $o_1, o_2, \cdots, o_T$

Initialization: the tracking target's location o_0 in I_0 $\chi^L = empty$, $\Im = empty$, $H_a(\cdot) = 0$, $H_t(\cdot) = 0$, $H(\cdot) = 0$, the max size of $\Im$: $Threshold$;
Collect labeled samples in frame I_0 and add them into χ^L;
for $f = 1; f \leq F; f++$ **do**
  Train $H_T(\cdot)$ on χ^L using AdaBoost;
  for $i = 1; i \leq |\Im|; i++$ **do**
    Get the i^{th} sample $< z_i, \ell_i >$ from $\Im$;
    Updating $H(\cdot)$ with $< z_i, \ell_i >$ according to Alg. 1;
  end
  for $i = 1; i \leq |\chi^L|; i++$ **do**
    Get the i^{th} sample $< x_i, y_i >$ from χ^L;
    Updating $H(\cdot)$ with $< x_i, y_i >$ according to Alg. 1;
  end
  $H^p(\cdot) = H(\cdot)$;
  Constructing unlabeled samples set U in I_f;
  for $i = 1; i \leq |U|; i++$ **do**
    Get the i^{th} sample u_i from U;
    Updating $H(\cdot)$ with u_i according to Alg. 1;
  end
  $o_f = \arg_{u_i} \max H(u_i)$;
  $\Im = \Im \cup \chi^L$;
  if $|\Im| > Threshold$ **then**
    Discard the oldest $|\Im| - Threshold$ samples
  end
  Update $H_a(\cdot)$ with χ^L using Online Boosting [8];
  Collect labeled samples in frame I_t and re-initialize them as χ^L;
end
return $\{o_1, o_2, \cdots, o_F\}$;

4.1. Implementation

We implement our tracking algorithm with C++ code. Some related details that readers will care about are described specifically.

Prior Distribution. Similar to [18], the prior distribution $r_{a,t}(x, y) = \frac{p(<x,y> \in \Im)}{p(<x,y> \in \chi^L)}$ in Eq. (5) is simply considered to be a constant $r_{a,t}$ and $r_{a,t} = 1$ in our experiments.



(a) (b)

Figure 4. (a) illustrates how to obtain labeled samples based on tracking result and (b) is about how to collect unlabeled data in frame I_{t+1}.

Feature. Although many features such as color, texture and HoG could be used for tracking, we select Haar based feature in our experiment because of its excellent performance reported in many works [22, 21]. Moreover, resorting to integral image, Haar feature value is very computationally efficient.

Weak Classifier. Provided Haar feature value v, the weak classifiers is defined as $h(v) = sign(p(v \in C_P|v) - p(v \in C_N|v))$. C_P and C_N denote object and background class respectively. We model prior distribution $p(v \in C_P)$ and $p(v \in C_N)$ with Gaussian distributions and update them during tracking. In our implementation, the number (N_w) of weak classifiers is 1000 and the number (M) of selectors is 100.

Memory. As described in Alg. 2, we need to store auxiliary samples for updating. To reduce the computational complexity, we store $h_i(z)$ $(i = 1, 2, \cdots, N_w)$ instead of z. This is because we perform updating with samples, we need to evaluate $h_i(\cdot)$. The maximal size of auxiliary samples is set as 1000 ($Threshold = 1000$). It is a compromise between tracking speed and accuracy.

Samples Collection. After we obtain tracking result in the current frame I_t, we collect labeled samples as shown in Fig. 4(a). In frame I_{t+1}, we collect unlabeled data using dense sampling method (overlap ratio is 0.99) in the search region and Fig. 4(b) provides a graphic representation.

5. Experimental Results and Analysis

In this section, we design experiments to demonstrate the superior properties of the proposed tracker. We compare with related works [8, 9, 2, 12] whose codes and parameters are provided on their websites. For simplicity, we refer online boosting tracker[8], semi-supervised online boosting tracker[9], online multiple instance learning tracker[2], tracking-learning-detection tracker[13], our online CovBoost tracker and semi-supervised online CovBoost tracker as **OBT**, **SSOBT**, **MILT**, **TLD**[1], **OCBT** and **SSOCBT**. All of those trackers except for **TLD** don't adapt to scale changes. We compare **OCBT**, **SSOCBT** with **OBT** and **SSOBT** respectively to demonstrate the effectiveness of our novel assumption. Secondly, we compare our tracker with four trackers on various test videos and prove that our tracker could alleviate "drift" to some extent.

[1]The learning part of **TLD** is P-N learning [12].

Figure 5. Qualitative comparison results of **SSOCBT**(in magenta), **OCBT**(in green), **OBT**(in red), **SSOBT**(in yellow) on "jump" sequences. The last image is the **SSD** of the face's image patches in adjacent frames.

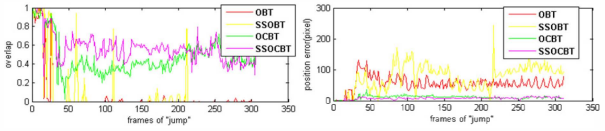


Figure 6. Quantitative comparison results of **SSOCBT**, **OCBT**, **OBT**, **SSOBT** on "jump" sequences.

5.1. The Effectiveness of Our Assumption

As in our experiments, the only difference between CovBoost Tracker and Boost Tracker is that their assumptions on samples' distribution are different. The test video is "jump" [12] which is recorded with a moving camera. The boy was skipping rope and his face is often blurry. In human eyes, the appearance of his face may not change significantly, but actually the differences between faces in adjacent frame is considerable as shown in the 4^{th} image in Fig. 5. However, **OBT** and **SSOBT** begin to drift soon after tracking because of motion blur. The incorrect updates severely degrade their performance. With the "Covaraite Shift" assumption on samples' distribution, **OCBT** and **SSOCBT** succeed in tracking the face through the whole video. Fig. 6 displays quantitative comparison results using Overlap-Criterion[2] and the position error computed as following.

$$overlap(g_f, o_f) = \frac{area\,(g_f \cap o_f)}{area\,(g_f \cup o_f)} \quad (15)$$

$$error(g_f, o_f) = \|center(g_f) - center(o_f)\|_2 \quad (16)$$

where g_t denotes ground truth, o_f is the tracking result and $\|\cdot\|_2$ is L_2 norm. In our experiment, the tracked target is denoted by a rectangle and g_f and o_f represent rectangles. From the result, we can see the improvement brought by our assumption is apparent and significant.

5.2. Comparison with Other Trackers

To show the superiority of **SSOBT** over other trackers, we perform experiments using **OBT**, **SSOBT**, **MILT**, **TLD**, **OCBT** and **SSOCBT** on a number of videos.

[2]http://www.pascal-network.org/challenges/VOC/voc2009

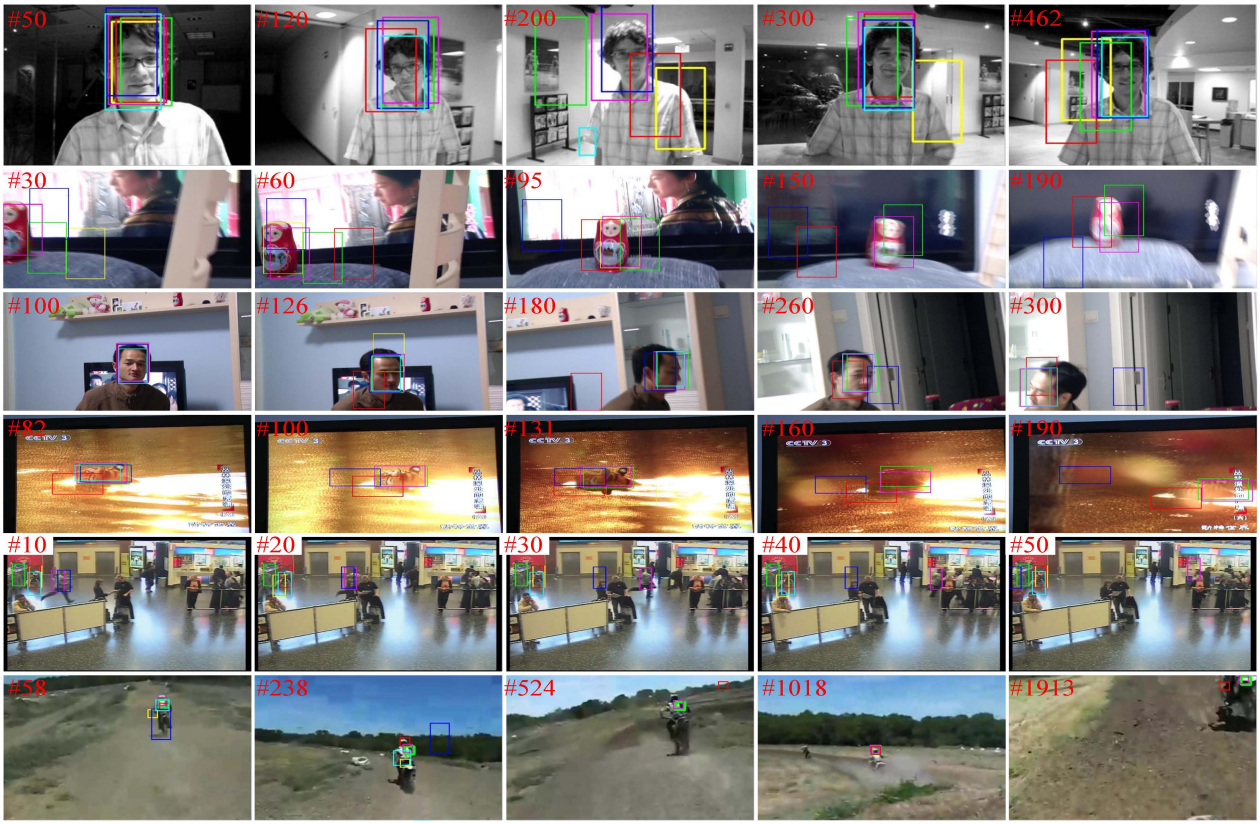
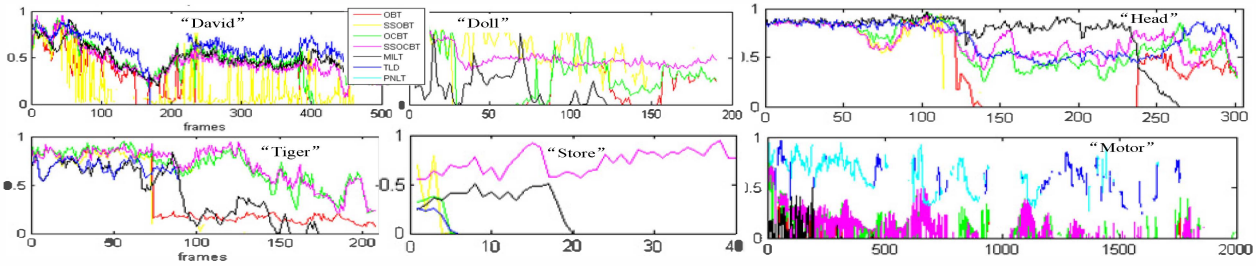


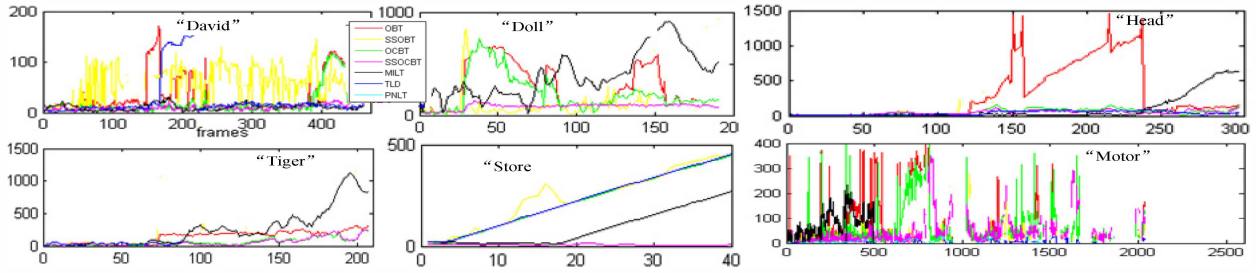
Figure 7. (Qualitative comparison results of **SSOCBT**(in magenta), **OCBT**(in green), **OBT**(in red), **SSOBT**(in yellow), **MILT**(in blue) and **TLD**(in cyan) on six test video sequences. In some frames, **SSOBT** or **TLD** doesn't provide tracking results, because it infers that the target disappears in those frames.

The first test video is "David" downloaded from `http://www.cs.utoronto.ca/~dross/ivt/`. In this video, a boy walks from a dark room to another one with light on. His pose, the appearances of his face and the background constantly change. The second video is "Doll" recorded by a moving camera taking a television as background. The television set is on and dynamic. So the background is continually changing and sometimes there are sudden variations. Meanwhile, due to the camera's movement, the target comes out fuzzy from time to time. The third video is "Head" recorded with a hand-held camera. The great challenge of this video is that it includes 180 degree out-of-plane rotation of the head and motion blur. The fourth test video is "Tiger" which is about a tiger. It is floating on the water. The glistening water reflecting the sun on the ripples and the shadows of the surroundings pose great challenge to track the tiger. Some time, the tiger is very blurred and appears very similar to the water. The fifth test video is "Store" which is from the video data of TRACKVID2010. In this video, a man runs through the room. The appearance of the background varies drastically. The last video "Motor" used in [12] is about a motor bumping along the rough mountain road. The motor disappears sometimes and its scale changes greatly.

The qualitative results on the above six test videos are displayed in Fig. 7. We can see that **OCBT** performs better than **OBT**, because our novel assumption on training samples helps the tracker select more discriminative features. Through exploiting the structure information implied in unlabeled samples, **SSOCBT** makes further improvement. During the process of tracking, **SSOBT** and **TLD** infer that whether the target is present or not, so they don't provide tracking results if they think the target disappears. In some frames in Fig. 7, **SSOBT** and **TLD** could not provide tracking results because of their wrong inferences. Especially for "Doll" sequence, **TLD** judges that the target disappear in the 3^{rd} frame and never appear again. **SSOBT** performs re-detection when it loses the target, so it could recover from failure (e.g. frame 60 in "Doll"). Applying CovBoost, when the doll is blurry, **SSOCBT** could identify the doll and it succeeds in capturing it through the whole video. "Tiger" is a very good test video for illustrating the advantage of our tracker. The appearances of the tiger and water vary gradually. The other trackers lose the tiger soon. One reason is that the data distribution is changing while it is assumed to be drawn from the same distribution for those



(a) Overlap of tracking results and ground truth on test videos. x-coordinate: frames, y-coordinate: overlap.



(b) Position error (pixel) between tracking results and ground truth on test videos. x-coordinate: frames, y-coordinate: position error.

Figure 8. Quantitative results on test videos using different trackers. The tracking result on **PNLT** on "Motor" is provided on its website and we make comparison with it too.

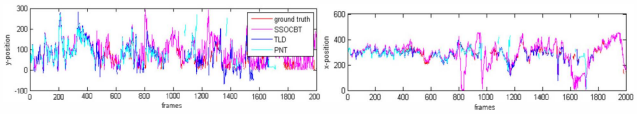


Figure 9. X and Y position of tracking results on "Motor" sequence using **SSOCBT**, **PNLT**[12] and **TLD**.

trackers. As we show in Fig. 2, semi-supervised CovBoost could generate a correct decision plane, and **SSOCBT** can adapt to dynamic changes and performs well. When distracter exists, **OBT**, **SSOBT**, **TLD** fail and the 10^{th} frame of "Store" is an example. **MILT** could insist on tracking the target for a few frames, but it loses the target when the target is close to the second distracter. **SSOCBT** performs better, but it loses the target in frame 50, because there is partial occlusion and the background is very similar to the target. The results on "Motor" prove that **SSOCBT** could cope with significant variations.

Given the ground truth g_f in every frame, we adopt overlap and position error quantitative comparisons, which are computed according to Eq. 15 and Eq. 16 respectively. This is because we think that overlap is very suitable for evaluating trackers which run with fixed scale. But overlap is very sensitive to scale, so if compare with **TLD**, position error is more fair. This is because, **TLD** is scale adaptive while the other trackers are not. This severely affects the overlap criterion. As position error is less sensitive to scale, we use it for quantitative comparison too. We plot the overlap as well as the position error curves for the test videos in Fig. 8. Apparently, **SSOCBT** performs better than **OBT**, **SSOBT** and **MILT** measuring with either overlap or position error. Compared with **TLD**, **SSOCBT** outperforms on tracking targets with no scale changes, such as "Doll", "Tiger" and "Store". For "David" and "Head" sequences, **TLD**'s overlap curves are a little higher than that of **SSOCBT**, because it is scale adaptive. The overlap of our results on "Motor" is low because the scale of the motor changes greatly. However, the position error of our results is encouraging. Fig. 9 shows x-coordinate and y-coordinate of **SSOCBT**'s, **PNLT**'s and **TLD**'s results for detail comparison. We can see that x-coordinate and y-coordinate of **SSOCBT**'s results are concordant with ground truth well. However, **SSOCBT** could not cope with target's disappearance and reappearance, so its position error increases sharply when the target reappears. Table 1 shows the comparison results in terms of the number of successful frames (referred as NOSF) in which the target is tracked successfully[3] and the average position error of tracking results in those frames. As we can see, **SSOCBT** achieves the largest NOSF in all the test videos.

6. Conclusions

In this work, we treat tracking as an online binary classification problem. After providing the tracking result in the current frame, the samples collected in the previous frames are considered as auxiliary data and the samples obtained in the current frame are regarded as target data. We assume

[3]Target is considered to be tracked successfully if the overlap between tracking result and ground truth is greater than zero.

Table 1. The NOSF of every tracker on test videos and the average position errors of tracking results in successful frames. The larger NOSF is, the better the tracker is.

sequence	David		Doll		Head		Tiger		Store		Motor	
	NOSF	error	NOSF	error	NOSF	error	NOSF	error	NOSF	error	NOSF	error
OCBT	401	**10.49**	145	131.60	301	53.94	207	61.21	5	22.63	725	17.70
SSOCBT	**460**	10.86	**190**	87.74	**301**	45.51	**207**	58.88	**43**	**8.62**	**1093**	20.76
OBT	328	19.48	120	121.37	202	63.14	207	136.86	5	22.63	507	17.98
SSOBT	344	53.79	140	**71.29**	120	28.40	84	49.96	4	19.93	291	20.44
MILT	460	11.27	101	228.24	263	**27.47**	163	134.92	19	16.21	127	28.30
TLD	407	12.08	1	88.07	301	39.51	89	**31.30**	5	26.02	1083	**7.49**

that the target and the auxiliary data are under "Covariate Shift" instead of following the same distribution. Furthermore, we think that the unlabeled samples in the next frame which we want to classify could provide useful information. We form a strong classifier based on the above three kinds of samples using semi-supervised CovBoost algorithm. It can select effective features for distinguishing the target from the background. We develop an online updating algorithm for semi-supervised CovBoost, which makes our approach easily be used for tracking and greatly improves the tracking performance, especially when the target's appearance changes heavily or there is motion blur.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China: 61025011, 60833006, 61035001 and 61003165, in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by Beijing Natural Science Foundation: 4111003.

References

[1] S. Avidan. Ensemble tracking. In *CVPR*, 2005. 1

[2] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009. 1, 5

[3] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, (1):63–84, 1998. 1

[4] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. *T-PAMI*, 27(1):1631–1643, 2005. 1

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *T-PAMI*, pages 564–577, 2003. 1

[6] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, 1994. 2

[7] A. Goldberg, M. Li, and X. Zhu. Online manifold regularization: A new learning setting and empirical study. In *ECCV*, 2008. 2

[8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006. 1, 2, 4, 5

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 1, 3, 4, 5

[10] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, 2004. 4

[11] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *IJCV*, pages 5–28, 1998. 1

[12] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010. 1, 5, 6, 7

[13] Z. Kalal, K. Mikolajczyk, and J. Matas. Face-tld: Tracking-learning-detection applied to faces. In *ICIP*, 2010. 5

[14] C. Leistner, H. Grabner, and H. Bischof. Semi-supervised boosting using visual similarity learning. In *CVPR*, 2008. 4

[15] R. Liu, J. Cheng, and H. Lu. A robust boosting tracker with minimus error bound in a co-training framework. In *ICCV*, 2009. 1

[16] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *T-PAMI*, 31(11):2000–2014, 2008. 3

[17] L. Matthews, T. Ishikawa, and S. Baker. The template update problem. *T-PAMI*, 26(6):810–815, 2004. 2

[18] J. Pang, Q. Huang, S. Jiang, and Z. Wu. Transfer pedestrian detector towards view-adaptiveness and efficiency. In *ICCV workshop on Video-Oriented Object and Event Classification*, 2009. 2, 4

[19] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 2000. 2

[20] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *ICCV*, 2007. 1

[21] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 5

[22] J. Wang, X. Chen, and W. Gao. Online selectiong discriminative tracking features using particle filter. In *CVPR*, 2005. 1, 2, 5

[23] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *CVPR*, 2010. 2

[24] Q. Yu, T. Dinh, and G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, 2008. 1

[25] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised mulitple-instance boosting. In *CVPR*, 2010. 1, 3

[26] X. Zhu. Semi-supervised learning literature survey. *Computer Science Technical Report 1530, Univ. of Wisconsin*. 3