A Robust Boosting Tracker with Minimum Error Bound in a Co-Training Framework

Rong Liu, Jian Cheng, and Hanqing Lu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Science, Beijing, China, 100190
{rliu, jcheng, luhq}@nlpr.ia.ac.cn

Abstract

The varying object appearance and unlabeled data from new frames are always the challenging problem in object tracking. Recently machine learning methods are widely applied to tracking, and some online and semi-supervised algorithms are developed to handle these difficulties. In this paper, we consider tracking as a classification problem and present a novel tracking method based on boosting in a co-training framework. The proposed tracker can be online updated and boosted with multi-view weak hypothesis. The most important contribution of this paper is that we find a boosting error upper bound in a co-training framework to guide the novel tracker construction. In theory, the proposed tracking method is proved to minimize this error bound. In experiments, the accuracy rate of foreground/background classification and the tracking results are both served as evaluation metrics. Experimental results show good performance of proposed novel tracker on challenging sequences.

1. Introduction

Visual tracking is an interesting but challenging problem in computer vision. Traditional methods [3] [9] treat tracking as a matching problem through building visual appearance models for the object. In recent years, a promising trend in tracking literature is treating object tracking as a classification problem using machine learning approach. Instead of constructing a complex model to describe the object, these classification methods seek a decision boundary that can best separate the object and the background. In this way, some challenging problems in tracking, such as background clutter and object dynamics, are naturally solved.

In [11], a classifier for distinguishing objects from background is online built and adaptively updated based on incremental LDA. However, LDA is a well know baseline classification learning method thus limits the performance improving of the tracker. In [1], a support vector tracker is constructed using support vector machine as the classifier. The off-line learning in this work decreases the tracker's adaptivity to complex object appearance change and background clutter. Ensemble tracking [2] using multiple features to distinguish object and background has superiority over the above methods. However, its classification just on pixels is easy to lost many structure information of the object appearance and lead to weakening the final tracker. Online boosting tracker [8] extracts region patches instead of pixels as samples. The seminal work is proposed by Oza *et al.* in [13]. Grabner *et al.* improve this idea through solving the feature selection task and utilize it to handle tracking problem. The powerful function of multiple features selection and weak classifiers ensemble in Grabner's online boosting lead to a more robust tracker.

The main problem of above methods is the self-training process which use the classification results to update the classifier itself. In self-training, classification mistakes reinforce themselves and the algorithm is not robust to outliers. To handle this problem, co-training [4] is a better option. It is a popular semi-supervised learning method and achieves success in many applications [5] [16]. Co-Tracking [15] online learns two independent SVM trackers in a co-training framework and improves each individual classifier using the information from other features. However, feature independence limits co-tracking to select more features to further improve the tracker performance.

To solve these problems, we present a novel tracker by using boosting learning in a co-training framework, which is online, multiple features based and semi-supervised. Some related works also combine boosting and co-training to construct learning approach, such as the method in [10]. However, co-training just on the final strong hypothesis as this method does is only a superficial combination of boosting and co-training, and how to design the combination strategy remains a problem deserving research. In this paper, a derived boosting error bound in a co-training framework is served as the theoretical guidance for the combi-

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

nation strategy. The proposed tracking method uses co-training to online learn each weak hypothesis in boosting instead of just the final strong hypothesis, and is proved to be minimizing the derived error bound in theory.

In the experiments, some challenging tracking sequences are captured. The accuracy rate of foreground/background classification and the tracking results are both served as evaluation metrics. Off-line boosting tracker, online boosting tracker [8] and co-tracker [15] are all involved in comparison with our method in the experiments. The experimental results show that our method outperforms the other three methods w.r.t the above metrics.

2. Related work

Boosting and co-training are two key components of our approach. Related works have proved that the minimization of training error upper bound is guaranteed in their frameworks. In this section, we briefly introduce some works about the error bound analysis for adaboost and co-training.

2.1. Error upper bound of adaboost

The classifier of adaboost is an ensemble of several weak hypothesis:

$$H(x) = \sum\nolimits_{t=1}^{T} \alpha_t h_t(x) \tag{1}$$

where the $H(x)$ is the ensemble strong hypothesis and its classification result is $sign(H(x))$. the $h_t(x)$ is the tth weak hypothesis to be learned and the α_t is the corresponding voting weight.

In [14], Schapire and Singer show that the training error of adaboost is upper bounded by

$$\frac{1}{n}\sum_{i=1}^{n} \exp(-y_i H(x_i)) = \prod_{t} Z_t \tag{2}$$

where the x_i is the ith sample in training and the y_i is the corresponding class label. The n is the training sample number and the Z_t is a normalization factor which is the weight sum of all the samples after the tth weak hypothesis training. Through minimizing Z_t in each weak hypothesis learning, adaboost decreases the whole error upper bound of itself. The Z_t can be expressed by:

$$Z_t = \sum_{i=1}^{n} D_t(i) \exp(-y_i \alpha_t h_t(x_i)) \tag{3}$$

where the $D_t(i)$ is the normalized weight of the ith sample in the tth weak hypothesis training.

2.2. Error upper bound of co-training

The theoretical analysis of co-training error bound first appear in [7] and then a research on the essence of co-training using graphical model is done in [4]. Based on the previous work, an Bayesian co-training error bound is derived in [16].

Our analysis is mainly based on the work in [7]. It proves that PAC-style guarantees that if (a) sample size are large, (b) the different views are conditionally independent given the class label, and (c) the classification decisions based on multiple views largely agree with each other, then *with high probability the misclassification rate is upper bounded by the rate of disagreement between the classifiers based on each view*. This property is exactly mentioned in [16]. On the assumption that there is no abstained samples to each view, the above error bound can be approximately expressed as follows:

$$P(h_j \neq l \,|y = l\,) \leq P(h_j \neq l \,|h_{3-j} = l\,) \tag{4}$$

where the $l \in \{-1, +1\}$ is a label and $y \in \{-1, +1\}$ is the real label. The $j \in \{1, 2\}$ is the index of the view and the h_j is the classifier based on the jth view.

3. The proposed tracking method

In this section, we propose a novel tracking method based on boosting learning in a co-training framework to treat tracking as an online semi-supervised ensemble learning problem. The proposed tracker is initialized with some labeled frames based on off-line boosting algorithm, and then updates in each new frame using new predicted results as unlabeled data. To treat unlabeled data in the update process, two independent views are adopted to describe each sample in a co-training framework. Such multi-view ensemble tracker can be expressed as below:

$$F(x) = \sum_{t}\sum_{j=1}^{2} \alpha_{t,j} h_{t,j}(x) \tag{5}$$

where the x is the sample patch in frames. The $h_{t,j}$ is the tth weak hypothesis in the jth view, and the $\alpha_{t,j}$ is the corresponding voting weight. The $F(x)$ can be considered as a confidence value for each sample patch, and is used to construct a confidence map in each frame. To get the current object position, tracking window initialized by previous tracking result is shifted to the best possible position of the confidence map.

To construct such a tracker described in Eq. (5), online hypothesis update, weak hypothesis design and samples extraction are three key factors.

3.1. Online hypothesis update

The proposed online hypothesis update approach is an online semi-supervised ensemble learning algorithm. The overall principle of this algorithm is depicted in Figure 1 and in Algorithm 1.

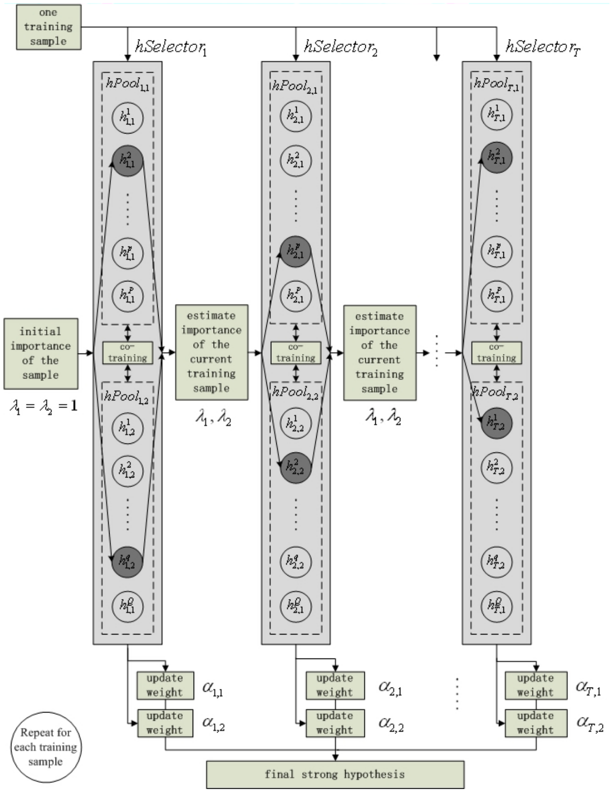

Figure 1. The proposed online semi-supervised learning framework

As shown in Figure 1, we select each weak hypothesis in two independent feature pools which represent independent sample views in a co-training framework. In Figure 1 and Algorithm 1, the $h_{t,j}^p$ is the pth weak hypothesis in the jth pool selected by the tth selector and the $h_{t,j}$ is the final selected weak hypothesis of the jth pool using the tth selector. The K is the iteration times of co-training in each selector and the $h_{t,j,k}$ is the correponding selected weak hypothesis in the kth co-training iteration. The $\lambda_{t,j,+}^p$ and the $\lambda_{t,j,-}^p$ are respectively the weighted correct rate and wrong rate of the pth weak hypothesis in the jth pool selected by the tth selector. The λ_j is the weight of new sample in the jth pool. The $\alpha_{t,j}$ denotes the voting weight of the tth selected weak hypothesis from the jth pool. The u_i is the label of the ith sample. For the labeled sample, the u_i is the real label. For the unlabeled sample, the u_i is the pseudo-label noted by independent views.

The proposed algorithm is an improvement on the online boosting described in [8]. Compared to the online boosting, the main characteristics of our method lie in four aspects: 1) input new sample is considered as unlabeled; 2) both parallel online boosting are performed simultaneously using weak hypothesis $h_{t,1}$ and $h_{t,2}$ from different feature pools; 3) co-training is performed to select weak hypothesis between feature pools in each selector; 4) the final strong hypothesis is a linear combination of strong classifiers training based on each independent view.

Algorithm 1 The proposed online semi-supervised ensemble learning approach for tracking hypothesis update

```
Input: unlabeled example $\langle x, \bullet \rangle$
Input: previous hypothesis $\tilde{h}_{t,j}^p$ $\tilde{h}_{t,j}$ for $\forall j, t, p$
Initialize: $\forall j, t, p : \lambda_j = 1$
$\lambda_{t,j,+}^p = \sum_{i:\tilde{h}_{l,j}^p = u_i} \exp(\sum_{l=1}^{t} -u_i \alpha_{l,j} \tilde{h}_{l,j}(x_i))$
$\lambda_{t,j,-}^p = \sum_{i:\tilde{h}_{l,j}^p \neq u_i} \exp(\sum_{l=1}^{t} -u_i \alpha_{l,j} \tilde{h}_{l,j}(x_i))$
// for all weak hypothesis
for $t = 1, \ldots, T$ do
  initialize $\forall j : h_{t,j,0} = \tilde{h}_{t,j}$
  // co-training iteration for both views
  for $k = 1, \ldots, K$ and $j = 1, 2$ do
    $u = sign(h_{t,3-j,k-1}(x))$ // set pseudo-label
    $\forall p : h_{t,j}^p = update(\tilde{h}_{t,j}^p, \langle x, u \rangle, \lambda_j)$ // update
    if $h_{t,j}^p(x) = u$ then
      $\hat{\lambda}_{t,j,+}^p = \lambda_{t,j,+}^p + \lambda_j$
    else
      $\hat{\lambda}_{t,j,-}^p = \lambda_{t,j,-}^p + \lambda_j$
    end if
    // select hypothesis $h_{t,j,k}$ with the lowest error
    $\forall p : e_{t,j}^p = \frac{\hat{\lambda}_{t,j,-}^p}{\hat{\lambda}_{t,j,+}^p + \hat{\lambda}_{t,j,-}^p}$
    $e_{t,j} = e_{t,j}^m$, $m = \arg\min_p(e_{t,j}^p)$
  end for
  $h_{t,j} = h_{t,j,K}$ // output selected hypothesis $h_{t,j}$
  $\alpha_{t,j} = \frac{1}{2} \ln \left( \frac{1-e_{t,j}}{e_{t,j}} \right)$ // calculate voting weight
  // update sample weight for $\forall j$
  if $h_{t,j} = u$ then
    $\lambda_j = \lambda_j \cdot \frac{1}{2(1-e_{t,j})}$
  else
    $\lambda_j = \lambda_j \cdot \frac{1}{2e_{t,j}}$
  end if
end for
```

3.2. Hypothesis design and samples extraction

The overall principle of samples extraction and weak hypothesis design is depicted in Figure 2.

For the weak hypothesis design, online update is a necessary condition to the weak hypothesis in Algorithm 1. As described in [8], a large number of hypothesis which meet this condition can be selected such as Bayesian hypothesis, Haar Wavelets based hypothesis and so on. In our algorithm described in Figure 1, all the weak hypothesis in the selector are divided into two classes to build different feature pools, and the hypothesis from different pools must be guaranteed

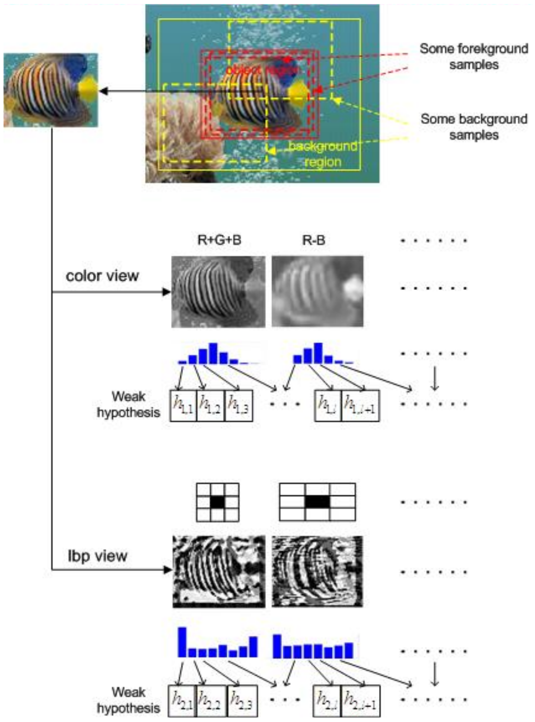

Figure 2. Process of weak hypothesis design

independent. As most of tracking algorithms, our feature pools are main based on color and texture features. The weak hypothesis is designed based on each bin of the histogram using Bayesian decision criterion. For the color features, many kinds of combination of camera R, G, B pixel values is used to build color histograms which is proved to be effective in [6]. For the texture features, several different LBP [12] construction in gray image is adopted to build LBP histograms.

For samples extraction, we assume that the object has already been tracked in the current frame and therefore have an initial object region which is a few pixels enlarged to the real object size. The tracker training uses the patches in the object region as positive samples and patches in the local neighborhood as negative samples.

4. Error analysis

In this section, we give the error analysis of the proposed algorithm described in Algorithm 1 and further prove that it minimize the error upper bound in theory.

4.1. Error upper bound

Without loss of generality, we give three assumptions for the following deduction: 1) the training sample is large scale; 2) the different views are conditionally independent given the class label; 3) the classification decisions based on multiple views largely agree with each other.

In the co-training framework, classifiers based on each independent views are trained at the same time, and then naturally combined together to give a final hypothesis. In adaboost, this final hypothesis can be denoted as below:

$$F(x) = \sum\nolimits_j H_j(x) \tag{6}$$

where the $H_j(x)$ is the strong classifier based on the jth view described in Eq.(1). In supervised learning, the error bound of $F(x)$ is derived as below.

Theorem 1 *Assuming the notation of Eq.(2) and Eq.(6), the following bound holds on the training error of $F(x)$ in supervised leaning:*

$$\frac{1}{n}|\{i : sign(F(x_i)) \neq y_i\}| \leq \frac{1}{J}\sum_{j=1}^{J}(\prod_{t=1}^{T} Z_{t,j})$$

where the j is the index of the view.

Proof For the jth view, we have the following normalized weight of the ith sample:

$$D_{T+1,j}(i) = \frac{\exp(-y_i H_j(x_i))}{n \prod_t Z_{t,j}} \tag{7}$$

Moreover, if $sign(F(x_i)) \neq y_i$ then $y_i F(x_i) \leq 0$ implying that $\exp(-y_i F(x_i)) \geq 1$ and $[\exp(-y_i F(x_i))]^{\frac{1}{J}} \geq 1$. Thus,

$$\begin{aligned}
[\![sign(F(x_i) \neq y_i)]\!] &\leq [\exp(-y_i F(x_i))]^{\frac{1}{J}} \\
&= [\exp(-y_i \sum_{j=1}^{J} H_j(x_i))]^{\frac{1}{J}} \\
&= \prod_{j=1}^{J} [\exp(-y_i H_j(x_i))]^{\frac{1}{J}} \\
&\leq \frac{1}{J}\sum_{j=1}^{J} \exp(-y_i H_j(x_i)) \quad (8)
\end{aligned}$$

where the $[\![\bullet]\!]$ is a bool function. Combining Eqs.(7) and (8) gives the stated bound on training error since

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}[\![sign(F(x_i) \neq y_i)]\!] \\
&\leq \frac{1}{n}\sum_{i=1}^{n}[\frac{1}{J}\sum_{j=1}^{J}\exp(-y_i H_j(x_i))] \\
&= \frac{1}{n}\sum_{i=1}^{n}[\frac{1}{J}\sum_{j=1}^{J}[n(\prod\nolimits_t Z_{t,j})D_{T+1,j}(i)]] \\
&= \frac{1}{J}\sum_{j=1}^{J}[(\prod\nolimits_t Z_{t,j})\sum_{i=1}^{n}D_{T+1,j}(i)] \\
&= \frac{1}{J}\sum_{j=1}^{J}(\prod\nolimits_t Z_{t,j}) \quad \blacksquare \qquad (9)
\end{aligned}$$

In semi-supervised learning, some samples are labeled whereas the others are unlabeled, so the bound in Theorem 1 can not be suitable. Assuming that the $\{x_i\,|i = 1, \ldots, m\}$ are the labeled samples and the $\{x_i\,|i = m+1, \ldots, n\}$ are the unlabeled samples, we deduce an error upper bound shown in Theorem 2 for semi-supervised learning in adaboost framework. To simplify the analysis, our research is restricted to two views, i.e. $J = 2$ in Theorem 1.

Theorem 2 *Assuming the three assumptions shown in the first paragraph of this subsection come true, the following bound holds on the training error of $F(x)$ in semi-supervised leaning:*

$$\begin{aligned}
&\frac{1}{n}\,|\{i : sign(F(x_i)) \neq y_i\}| \\
&\leq \frac{1}{2n}\{\sum_{i=1}^{m}\exp(-y_i H_1(x_i)) + \sum_{i=1}^{m}\exp(-y_i H_2(x_i)) \\
&+ \sum_{i=m+1}^{n} \exp(\sum_{t=1}^{T} -sign(h_{t,2}(x_i)) \cdot \alpha_{t,1} h_{t,1}(x_i)) \\
&+ \sum_{i=m+1}^{n} \exp(\sum_{t=1}^{T} -sign(h_{t,1}(x_i)) \cdot \alpha_{t,2} h_{t,2}(x_i))\}
\end{aligned}$$

Proof Combining the conclusion of Theorem 1 and Eq.(2), the error bound of $F(x)$ can be expressed as below:

$$\begin{aligned}
&\frac{1}{n}\,|\{i : sign(F(x_i)) \neq y_i\}| \qquad (10)\\
&\leq \frac{1}{2n}\{\sum_{i=1}^{n}\exp(-y_i H_1(x_i)) + \sum_{i=1}^{n}\exp(-y_i H_2(x_i))\}
\end{aligned}$$

In semi-supervised learning, the $\{x_i\,|i = 1, \ldots, m\}$ are the labeled samples and the $\{x_i\,|i = m+1, \ldots, n\}$ are the unlabeled samples. The Eq.(11) can be expressed as follows:

$$\begin{aligned}
&\frac{1}{n}\,|\{i : sign(F(x_i)) \neq y_i\}| \qquad (11)\\
&\leq \frac{1}{2n}\{\sum_{i=1}^{m}\exp(-y_i H_1(x_i)) + \sum_{i=1}^{m}\exp(-y_i H_2(x_i)) \\
&+ \sum_{i=m+1}^{n}\exp(-y_i H_1(x_i)) + \sum_{i=m+1}^{n}\exp(-y_i H_2(x_i))\}
\end{aligned}$$

The $\{y_i\,|i = m+1, \ldots, n\}$ in Eq.(11) is unknown yet. Combining the Eqs.(3)(4)(7), the third term on the right side of Eq.(11) can be transformed as below: ($\sum\limits_{i=m+1}^{n} D_{T+1,1}(i)$ is denoted as D)

$$\begin{aligned}
&\sum_{i=m+1}^{n}\exp(-y_i H_1(x_i)) = \sum_{i=m+1}^{n}(\prod_{t=1}^{T} Z_{t,1}) D_{T+1,1}(i) \qquad (12)\\
&= D \cdot \prod_{t=1}^{T} Z_{t,1} = D \cdot \prod_{t=1}^{T}[\sum_{i=1}^{n} D_{t,1}(i)\exp(-y_i\alpha_{t,1}h_{t,1}(x_i))] \\
&= D \cdot \prod_{t=1}^{T}[W_{t,1,+}\cdot\exp(-\alpha_{t,1}) + W_{t,1,-}\cdot\exp(\alpha_{t,1})] \\
&= D \cdot \prod_{t=1}^{T}[\exp(-\alpha_{t,1}) + W_{t,1,-}\cdot(\exp(\alpha_{t,1}) - \exp(-\alpha_{t,1}))]
\end{aligned}$$

where $W_{t,1,+} = \sum_{i:h_{t,1}(x_i)=y_i} D_{t,1}(i)$ and $W_{t,1,-} = \sum_{i:h_{t,1}(x_i)\neq y_i} D_{t,1}(i)$

In the final expression of Eq.(12), only the $W_{t,1,-}$ is related with the y_i. In the form of $W_{t,1,-}$, it is the weighted error rate in the tth weak hypothesis training, e.i. $W_{t,1,-} = P(h_{t,1} \neq l\,|y = l\,)$.

Assuming the three assumptions given in the first paragraph of this subsection come true, the $P(h_{t,1} \neq l\,|y = l\,)$ is upper bounded by $P(h_{t,1} \neq l\,|h_{t,2} = l\,)$ based on the Eq.(4). Since the Eq.(12) is a increasing function of $W_{t,1,-}$, we can use $P(h_{t,1} \neq l\,|h_{t,2} = l\,)$ to replace $W_{t,1,-}$ e.i. $P(h_{t,1} \neq l\,|y = l\,)$ to get the upper bound of Eq.(12). This replacement is equal to replace the y_i with $sign(h_{t,2}(x_i))$. Through similar transformation, the upper bound of the fourth term on the right side of Eq.(11) can be also obtained. They are expressed as below: (for $j = \{1, 2\}$)

$$\begin{aligned}
&\sum_{i=m+1}^{n}\exp(-y_i H_j(x_i)) \qquad (13)\\
&\leq \sum_{i=m+1}^{n}\exp(\sum_{t=1}^{T} -sign(h_{t,3-j}(x_i))\cdot\alpha_t h_{t,j}(x_i))
\end{aligned}$$

Through combining Eqs.(11) and (13), the Theorem 2 is proved. ■

4.2. Error upper bound minimization

In this subsection, we minimize the error bound presented in Theorem 2, and use it to guide the strong hypothesis construction. For the Lth weak hypothesis learning, the weak hypothesis $h_{t,j}$ and corresponding weight $\alpha_{t,j}$ from $t = 1$ to $t = L-1$ are already known, and the objective is to solve $h_{L,j}$ and $\alpha_{L,j}$. This can be described as minimizing the function below:

$$\begin{aligned}
&B(h_{L,1}, h_{L,2}, \alpha_{L,1}, \alpha_{L,2}) \qquad (14)\\
&= \sum_{j=1}^{2}\sum_{i=1}^{n} D_{L,j}(i)\exp(-u_i\alpha_{L,j}h_{L,j}(x_i))
\end{aligned}$$

where

$$u_i = \begin{cases} y_i & i = 1 \ldots m \\ sign(h_{L,3-j}(x_i)) & i = m+1 \ldots n \end{cases}$$

$$D_{L,j}(i\,|i = 1 \ldots m) = \prod_{t=1}^{L-1} \exp(-y_i \alpha_{t,j} h_{t,j}(x_i))$$

$$D_{L,j}(i\,|i = m+1 \ldots n)$$

$$= \prod_{t=1}^{L-1} \exp(-sign(h_{t,3-j}(x_i)) \cdot \alpha_{t,1} h_{t,j}(x_i))$$

To simplify the solving process, an iterative minimization which fixes $h_{L,3-j}$ to solve $h_{L,j}$ is adopted instead of directly minimizing the Eq.(14). It is similar to the iteration optimization in the co-training framework. If the $h_{L,3-j}$ is fixed, minimization of Eq.(14) is transformed to minimize the function below:

$$B(h_{L,j}, \alpha_{L,j}) = \sum_{i=1}^{n} D_{L,j}(i) \exp(-u_i \alpha_{L,j} h_{L,j}(x_i)) \tag{15}$$

It can easily be verified that $B(h_{L,j}, \alpha_{L,j})$ is minimized when

$$\alpha_{L,j} = \frac{1}{2} \ln \left(\frac{W_{L,j,+}}{W_{L,j,-}} \right) \tag{16}$$

where $W_{L,j,+} = \sum_{i:h_{L,j}(x_i)=u_i} D_{L,j}(i)$ and $W_{L,j,-} = \sum_{i:h_{L,j}(x_i)\neq u_i} D_{L,j}(i)$

For this setting of α, we have

$$B(h_{L,j}) = 2\sqrt{W_{L,j,+} W_{L,j,-}} \tag{17}$$

We select $h_{L,j}$ to minimize the $B(h_{L,j})$ in Eq.(17). It is equal to minimize the weight error rate $W_{L,j,-}$.

The weight updates of the samples are shown in Eq.(18).

$$\begin{aligned} &D_{L+1,j}(i\,|i = 1 \ldots m) = D_{L,j}(i\,|i = 1 \ldots m) \qquad (18) \\ &\cdot \exp(-y_i \alpha_{L,j} h_{L,j}(x_i)) \\ &D_{L+1,j}(i\,|i = m+1 \ldots n) = D_{L,j}(i\,|i = m+1 \ldots n) \\ &\cdot \exp(-sign(h_{L,3-j}(x_i)) \alpha_{L,j} h_{L,j}(x_i)) \end{aligned}$$

The final strong hypothesis described in Eq.(6) is obtained by linear combination of all the weak hypothesis in each views, i.e. $F(x) = \sum_{t} \sum_{j=1}^{2} \alpha_{t,j} h_{t,j}(x)$.

The above process for error upper bound minimization can be concluded in Algorithm 2.

Based on the idea of online boosting in [8], The proposed tracking hypothesis update approach in Algorithm 1 can be easily derived from Algorithm 2. So the Algorithm 1 is naturally proved to also minimize the error bound in Theorem 2, which is the error upper bound of boosting in a co-training framework.

Algorithm 2 Strong hypothesis construction with minimum error bound in Theorem 2

```
**Input:** labeled examples $\{\langle x_i, y_i \rangle\,|i = 1, \ldots, m\}$ and unlabeled examples $\{\langle x_i, \bullet \rangle\,|i = m+1, \ldots, n\}$
**Initialize:** $\forall i, j : D_{1,j}(i) = 1/n$
// for all weak hypothesis
for $t = 1, \ldots, T$ do
    initialize $\forall j : h_{t,j,0}$ using only labeled data.
    // co-training iteration for both views
    for $k = 1, \ldots, K$ and $j = 1, 2$ do
        // set pseudo-labels
        $u_i = \begin{cases} y_i & i = 1 \ldots m \\ sign(h_{t,3-j,k-1}(x_i)) & i = m+1 \ldots n \end{cases}$
        // choose hypothesis $h_{t,j,k}$ with the lowest error
        $W_{t,j,-} = \sum_{i:h_{t,j,k}(x_i)\neq u_i} D_{t,j}(i)$
    end for
    $h_{t,j} = h_{t,j,K}$ // output selected hypothesis $h_{t,j}$
    $\alpha_{t,j} = \frac{1}{2} \ln \left( \frac{1-W_{t,j,-}}{W_{t,j,-}} \right)$ // calculate voting weight
    // update sample weights for $\forall i, j$
    $D_{t+1,j}(i) = \frac{D_{t,j}(i) \cdot \exp(-u_i \alpha_{t,j} h_{t,j}(x_i))}{Z_{t,j}}$
    where $Z_{t,j} = \sum_i D_{t,j}(i) \cdot \exp(-u_i \alpha_{t,j} h_{t,j}(x_i))$
end for
// output final strong hypothesis
$F(x) = \sum_{t=1}^{T} \sum_{j=1}^{2} \alpha_{t,j} h_{t,j}(x)$
```

5. Experiments

In order to verify the proposed method, comparison experiments are performed on Face sequence, Car sequence, and Cup sequence. Some frames of them are shown in Figure 4.

In the experiments, tracking results and accuracy rate of foreground/background classification are both served as evaluation metrics. The experimental comparison is performed among four approaches: 1) *baseline*, which only uses labeled samples to train a off-line Adaboost classifier without update; 2) *online boosting* [8], which is initialized with the labeled data and online updated with unlabeled data through self-training; 3) *boosting co-tracking*, which uses the approach described in [15], while replaces the online SVM classifier with the online boosting classifier; 4) *our method*, which is described in Algorithm 1.

For classification comparison, 100 continuous and challenge frames in a video are used to generate training and testing data. Samples from earlier 10 frames are served as labeled and from other 90 frames are served as unlabeled.

For tracking comparison, all the algorithms begin with 10 manual labeled frames, and the tracking results in each new frame are used to locate the foreground and background regions to extract the samples as unlabeled data for online updating the trackers.

The approach for extracting samples and features is described in section 3. As most of tracking algorithms, we also adopt color features and texture features to describe the object.

For the color features, the following combinations [6] are used to construct color histogram.

$$C \equiv \{w_1 R + w_2 G + w_3 B \,|w_1, w_2, w_3 \in [-2, -1, 0, 1, 2]\}$$

Through pruning redundant coefficient, we are left with 49 combinations. All the color combinations are normalized into $64(4 \times 4 \times 4)$ dimensional histograms and each bin of these histograms is added into the color feature pool.

For the texture features, 25 kinds of LBP blocks with different sizes are selected to generate the corresponding histograms. The LBP histogram are also normalized into $64(4 \times 4 \times 4)$ dimensions and each bin of them is added into the LBP feature pool.

5.1. Comparison of classification

The results of classification comparison in face tracking sequence and car tracking sequence are shown in Figure 3.

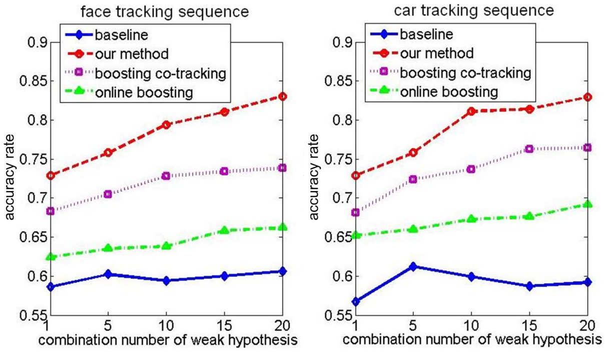

Figure 3. The classification accuracy rate comparison of *baseline* (blue), *online boosting* (green), *boosting co-tracking* (purple) and *our method* (red) in a face tracking sequence (left) and a car tracking sequence (right).

As shown in Figure 3, *baseline* is the worst approach since its off-line training with no update. Taking account of classifier online update help *online boosting* get a better performance. However, *Online boosting* using the classification results to update the classifier itself is not robust to outliers. So its performance is worse than *boosting co-tracking* which use co-training to replace with the self-training. Compared to *boosting co-tracking*, *our method* does co-training in each weak hypothesis, and plays a better performance in experiments. In conclusion, *our method* which is an entitative combination of boosting and co-training based on the error upper bound minimization is shown the best performance in classification comparison experiments.

In the comparison of classification, we also use various combination number of boosting weak hypothesis to make a comparison. The results in Figure 3 show that the proposed method gain the most improvement with increasing of combination. This indicates that *our method* is a suitable combination between boosting and co-training.

5.2. Comparison of tracking

Figure 4 shows the results of tracking comparison.

In the face tracking sequence, face rotation, illumination change in the middle scene and clutter background are the challenging problem. *baseline approach* using off-line boosting without update is unstable after the face rotation in the earlier frame and loses track in frame #21. Since the very similar color features in the door, *online boosting* using self training fails to track the face in frame #86. After 142 frames tracking, only *our method* keeps correct.

In the car tracking sequence, complex illumination conditions are created by the shadow in the scene and a strong sunshine reflection happened in frame #46 is also a big challenge. *baseline approach* loses the car on the border between the illumination and the shadow in frame #12. When the strong sunshine reflection happened in frame #46, all the algorithms are affected and get incorrect labeled samples to train. For this reason, *online boosting* fails to track the car in the following frames. In the last frames, *boosting co-tracking* can reluctantly keep tracking whereas *our method* plays a better performance.

In the cup tracking sequence, the cup tracking is done in a dark room with faint light. illumination change leads to a tracking lose of *baseline approach* in frame #16. In frame #76, the cup rotation affect the performance of all the tracking algorithm. Only *our method* still keeps tacking after 92 frames.

The above experiments show a good tracking performance of *our method* compared to *baseline approach*, *online boosting* and *boosting co-tracking*.

6. Conclusions

In this paper we pose tracking as a online semi-supervised learning problem and present a robust tracker using online boosting and co-training framework. For guiding the design of our algorithm, we find an error upper bound for Adaboost in semi-supervised learning based on a co-training framework and prove it in theory. Our method is based on minimizing this error upper bound. Off-line boosting tracker, online-boosting tracker and boosting co-tracker are compared to our tracker. The solid theory support and convincible experimental results show that the proposed tracker is promising.

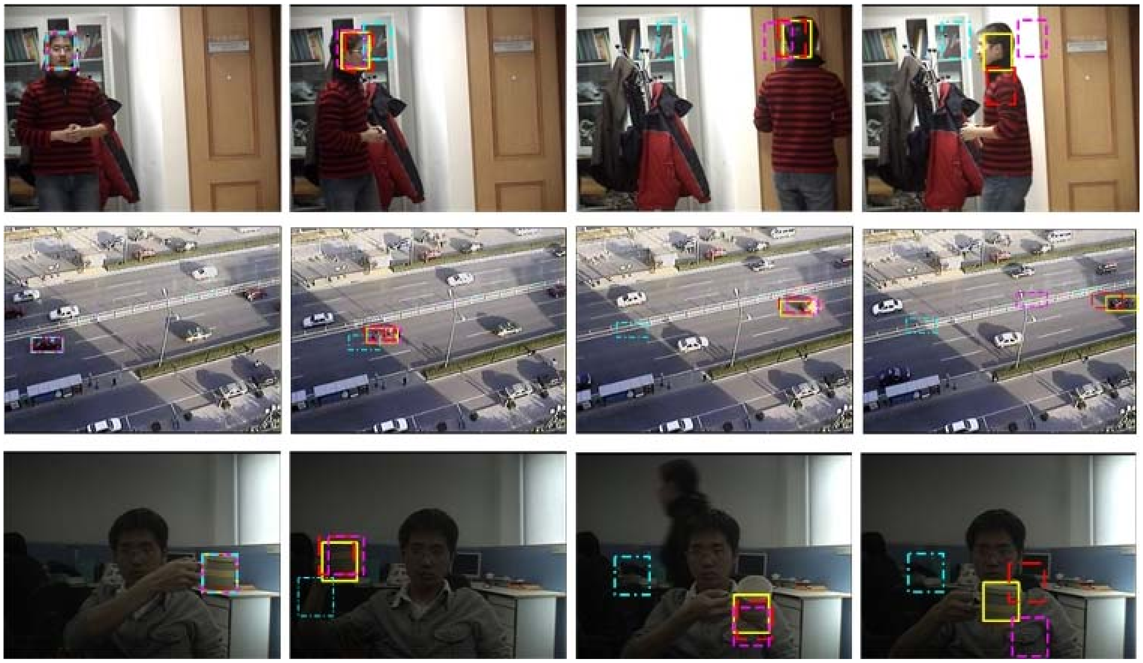

Figure 4. Tracking performance comparison of *baseline approach* (cyan rectangles), *online boosting* (purple rectangles), *boosting co-tracking* (red rectangles) and *our method* (yellow rectangles). The top row shows the comparison results on frames 1, 21, 86 and 142 from a face tracking sequence. The middle row shows the comparison results on frames 1, 12, 46 and 57 from a car tracking sequence. The bottom row shows the comparison results on frames 1, 16, 76 and 92 from a cup tracking sequence.

Acknowledgement

This work is partially supported by 973 Program (Project No. 2010CB327900 and 2010CB327905), National Natural Science Foundation of China (Grant No. 60605004 and 60833006) and Natural Science Foundation of Beijing (Grant No. 4072025).

References

[1] S. Avidan. Support vector tracking. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 26, pages 1064–1072, 2004. 1

[2] S. Avidan. Ensemble tracking. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 29, pages 261–271, 2005. 1

[3] M. Black. Eigentracking: Robust matching and tracking of articulated objects using a view-based rprenstation. In *International Journal of Computer Vision*, volume 26, pages 329–342, 1996. 1

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. pages 92–100. ACM, 1998. 1, 2

[5] J. Chan, I. Koprinska, and J. Poon. Co-training with a single natural feature set applied to email classification. *Proc. IEEE Int'l Conf. on Web Intelligence*, pages 586–589, 2004. 1

[6] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 27, pages 1631–1643, 2005. 4, 7

[7] S. Dasgupta, M. L. Littman, and D. Mcallester. Pac generalization bounds for co-training. In *NIPS*, volume 1, pages 375–382, 2001. 2

[8] H. Grabner and H. Bischof. Online boosting and vision. *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Vol.1:260–267, June 2006. 1, 2, 3, 6

[9] M. Isard and A. Blake. Condensation: conditional density propagation for visual tracking. In *International Journal of Computer Vision*, volume 29, pages 329–342, 1998. 1

[10] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using cotraining. *Proc. IEEE Int'l Conf. on Computer Vision*, Vol.1:626–633, Oct 2007. 1

[11] G. Li, D. Liang, Q. Huang, S. Jiang, and W. Gao. Object tracking using incremental 2d-lda learning and bayes inference. *Proc. IEEE Int'l Conf. on Image Processing*, pages 1568–1571, Oct 2008. 1

[12] P. Matti. Image analysis with local binary patterns. In *Image Analysis*, pages 115–118, 2005. 4

[13] N. Oza. Online bagging and boosting. *Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics*, Vol.3:2340–2345, Oct 2005. 1

[14] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Machine Learning*, volume 37, pages 297–336, 1999. 2

[15] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. *Proc. IEEE Int'l Conf. on Computer Vision*, pages 14–21, Oct 2007. 1, 2, 6

[16] S. Yu, B. Krishnapuram, R. Romer, H. Steck, and B. Rao. Bayesian co-training. In *NIPS*, volume 20, pages 1665–1672, 2008. 1, 2