Robust Tracking: Keeping Adaptivity but Refusing to Drift

Wenhui Dong[1, 2], Faliang Chang[1] and Zijian Zhao[1]

[1]*School of Control Science and Engineering, Shandong University, Jinan 250061, China*
[2]*Department of Physics, Dezhou University, Dezhou 253023, China*
dongwh@sdu.edu.cn, flchang@sdu.edu.cn, zhaozijian@sdu.edu.cn

Abstract

Tracking with a discriminative classifier becomes popular recently. The online updating makes it easy to adapt to target appearance variations. However, this also brings drifting problem. It's necessary to find a tracking method with strong adaptivity and anti-drifting ability. In this paper, an online semi-supervised boosting method is proposed at first, and based on it, we propose a novel tracking framework that treats samples differently when updating the classifier under different conditions. This tracking framework can significantly alleviate the drifting problem and keep adaptive enough to appearance variations. Experimental results on challenging videos show that our method can track accurately and robustly， and outperform many other state-of-the-art trackers.

Keywords: *Visual tracking, Discriminative classifier, Semi-supervised learning, Adaptivity, Drifting*

1. Introduction

Visual tracking is one of the key problems in computer vision. Its main task is to find the trace of the interesting object in each frame of the video. Although visual tracking has many practical applications [1, 2, 3], designing a robust and efficient tracking system is still a very challenging task. The tracker has to deal with many difficult situations which may occur in natural scenes, among them handling appearance variations of the target is most challenging [4]. There are two types of appearance variations: intrinsic and extrinsic. The intrinsic appearance variations include pose variation and shape deformation of the target whereas the extrinsic variations are due to the changes resulting from different illumination, camera motion, and partial occlusion. In order to handle such variations, the object model needs to be adjusted to the new circumstances from time to time. Thus online learning algorithms are essentially needed to incrementally update the appearance of the target [5]. Generally, there are many online learning based tracking algorithms. Among them, discriminative online learning methods are very popular recently.

Discriminative methods consider tracking as a binary classification task. The target identified by the user in the first frame is described by a set of features. Another set of features describes the background, and a binary classifier is trained to find the best decision boundary that can separate target from background. When the following frames come, the current classifier evaluates all possible positions in the search region to locate the target. Then the current target region and its surroundings are exacted as positive and negative samples respectively to update the classifier. Tian *et al.*, [6] present a tracking algorithm based on ensemble of online linear SVM classifiers. The SVM classifiers are updated during different periods with different historical information. Grabner *et al.*, [7] propose an online adaboost feature selection algorithm for tracking, which selects and maintains the best discriminative

features from a pool of feature candidates. Saffari *et al.*, [8] propose the online version of random forest algorithm based on an online decision tree growing procedure and use it as the online classifier for tracking. Generally speaking, discriminative methods are fast and yield good performance since the classification task is simple. However, the online adaptation also faces a key problem called drifting, which may finally lead to tracking failure. In order to solve this problem, Grabner *et al.*, [9] formulate tracking as a semi-supervised learning problem. The work aims to build an ensemble classifier utilizing the unlabeled samples. The loss function is built based on the similarity of labeled examples, labeled and unlabeled examples, and unlabeled examples respectively. Boosting is used to minimize the loss function. Tracking is then considered as a one-shot semi-supervised learning problem. Only samples in the first frame are labeled and all the samples in the following frames are considered as unlabeled. Then, the similarity of the unlabeled sample and the set of positive or negative samples are measured to obtain the pseudo-label of the unlabeled sample. Although this method has shown to be less susceptible to drifting and simultaneously more adaptive than an offline learner, it turns out that such an approach is still not adaptive enough for appearance variation.

A robust tracking method that has strong adaptivity and anti-drifting ability is proposed in this paper. There are two contributions in the method. First, we propose an online semi-supervised boosting method. Kullback-Leibler divergence between the prior probability of label distribution and the optimized model is used as the loss function for unlabeled samples. Then the loss term encourages the optimization procedure to find parameters that predict a similar label distribution on the unlabeled examples in online mode. Second, based on the online semi-supervised boosting, we propose a novel tracking framework that treats samples differently when updating the classifier under different conditions. This tracking framework can significantly alleviate the drifting problem and has enough adaptivity.

The reminder of the paper is organized as follows. In Section 2, we present our novel online semi-supervised boosting method. Section 3 is the detail of our novel tracking framework. Experimental results are shown in Section 4. Finally, in Section 5, we give some conclusions.

2. Online Semi-supervised Boosting

Inspired by the reference [10], an online semi-supervised boosting algorithm is proposed. Our goal is to obtain an additive model $F(x)=\sum_{t=1}^{T} f_t(x)$ in online mode. Where, $F(x)$ represents the strong classifier and $f_t(x)$ is the weak classifier.

2.1. Loss Function for Labeled and Unlabeled Samples

Given a set of samples $X = X_L \cup X_U$, X_L and X_U are labeled and unlabeled samples respectively. We define loss function for the semi-supervised boosting as:

$$L(F(x), X) = L_l(F(x), X_L) + \lambda L_u(F(x), X_U) \quad (1)$$

Where, $L_l(F(x), X_L)$ and $L_u(F(x), X_U)$ are loss functions for the labeled and unlabeled data respectively. λ is the contribution of the unlabeled data.

(1)Loss function for labeled samples

For the labeled samples, exponential function is used.

$$L_l(F(x), X_L) = \sum_{x \in X_L} e^{-yF(x)} \tag{2}$$

(2)Loss function for unlabeled samples

It is natural to think that if we have the prior distribution of the sample labels, then the classifier will be the optimal if it gives the same label distribution on the unlabeled samples. So the loss function for unlabeled samples can be defined as the distance between the prior label distribution and the condition probability given by the calculated model. As in reference [11], Kullback-Leibler divergence is used.

Suppose $\tilde{P}(y \mid x)$ is the prior probability, $P(y \mid x)$ is the condition probability of the calculated model, the loss function for the unlabeled data is defined as:

$$L_u(F(x), X_U) = \sum_{x \in X_U} e^{DKL(\tilde{P} \| P)} \tag{3}$$

Where, the function $DKL(\tilde{P} \| P) = \sum_y \tilde{P} \log \frac{\tilde{P}}{P} = \sum_y \tilde{P}(y \mid x) \log \tilde{P}(y \mid x) - \sum_y \tilde{P}(y \mid x) \log P(y \mid x)$ represents Kullback-Leibler divergence. Because the first term doesn't depend on the model and boosting can be viewed as additive logistic regression [11], thus the loss function can be written as:

$$\begin{aligned} L_u(F(x), X_U) &= \sum_{x \in X_U} e^{DKL(\tilde{P} \| P)} \approx \sum_{x \in X_U} e^{-\sum_y \tilde{P}(y|x) \log P(y|x)} \\ &= \sum_{x \in X_U} e^{[-\tilde{P}(y=1|x) \log \frac{e^{F(x)}}{e^{F(x)}+e^{-F(x)}} - (1-P(y=1|x)) \log \frac{e^{-F(x)}}{e^{F(x)}+e^{-F(x)}}]} \\ &= \sum_{x \in X_U} e^{-[(2\tilde{P}(y=1|x)-1)F(x) - \log(e^{F(x)}+e^{-F(x)})]} \end{aligned} \tag{4}$$

Define $\tilde{y} = 2\tilde{P}(y=1 \mid x) - 1,\ \tilde{y} \in [-1,1]$, then

$$L_u(F(x), X_U) = \sum_{x \in X_U} e^{-\tilde{y}F(x)} . (e^{F(x)} + e^{-F(x)}) = 2 \sum_{x \in X_U} e^{-\tilde{y}F(x)} \cosh(F(x)) \tag{5}$$

2.2. Learning based on Gradient Descent Principle

The overall loss function is:

$$L(F(x), X) = \sum_{x \in X_L} e^{-yF(x)} + \lambda \sum_{x \in X_U} e^{-\tilde{y}F(x)} \cosh(F(x)) \tag{6}$$

Then we can get the gradient of the loss function with respect to the current $F(x)$:

$$\begin{aligned} \frac{\partial L}{\partial F(x)} &= \sum_{x \in X_L} -ye^{-yF(x)} + \lambda \sum_{x \in X_U} -\tilde{y}e^{-\tilde{y}F(x)} \cosh(F(x)) + \lambda \sum_{x \in X_U} e^{-\tilde{y}F(x)} \sinh(F(x)) \\ &= \sum_{x \in X_L} -ye^{-yF(x)} + \lambda \sum_{x \in X_U} -e^{-\tilde{y}F(x)} (\tilde{y} \cosh(F(x)) - \sinh(F(x))) \end{aligned} \tag{7}$$

According to the gradient descent principle, the algorithm is looking for the weak classifier $f_t(x)$ at each step of boosting, which if added to the current $F(x)$ will result in an overall performance improvement. Thus, the boosting optimization problem for adding a weak classifier $f_t(x)$ at step t can be formulated as:

$$\begin{aligned} f_t(x) &= \arg\min_{f\in F} < \frac{\partial L}{\partial F(x)}, f(x) > \\ &= \arg\min_{f\in F}\{\sum_{x\in X_L} -ye^{-yF(x)} f(x) + \lambda \sum_{x\in X_U} -e^{-\tilde{y}F(x)}(\tilde{y}\cosh(F(x)) - \sinh(F(x)))f(x)\} \end{aligned} \tag{8}$$

Define $\omega_l = e^{-yF(x)}$, $\hat{y}_u = \tilde{y}\cosh(F(x)) - \sinh(F(x))$, $\omega_u = \lambda\left|\hat{y}_u\right| e^{-\tilde{y}F(x)}$,then the formula can be written as:

$$\begin{aligned} f_t(x) &= \arg\min_{f\in F}\{\sum_{x\in X_L} -y\omega_l f(x) + \sum_{x\in X_U} -sign(\hat{y}_u)\omega_u f(x)\} \\ &= \arg\min_{f\in F}\{\sum_{y\neq f(x)} \omega_l - \sum_{y=f(x)} \omega_l + \sum_{sign(\hat{y}_u)\neq f(x)} \omega_u - \sum_{sign(\hat{y}_u)=f(x)} \omega_u)\} \\ &= \arg\min_{f\in F}\{(\sum_{y\neq f(x)} \omega_l + \sum_{sign(\hat{y}_u)\neq f(x)} \omega_u) - (\sum_{y=f(x)} \omega_l + \sum_{sign(\hat{y}_u)=f(x)} \omega_u)\} \end{aligned} \tag{9}$$

If we consider ω_l as the weight of the labeled sample, $sign(\hat{y}_u)$ as the pseudo label and ω_u as the weight of the unlabeled sample, equation (9) means that in offline mode, the weak classifier we should choose in each step is the one with the minimum classification error in given weight distribution.

2.3. Online semi-supervised Boosting Algorithm

For the purpose of tracking, we need the online mode. In off-line case, all samples with a given weight distribution are used to select one weak classifier and the weight distribution is updated before choosing the next one. In online case, one sample is used to update all weak classifiers and we do not know the weight of each sample. In order to obtain an online semi-supervised boosting for tracking, there are three main questions to be answered: 1) How to get the number of training times for each sample to update the weak classifier? 2) How to choose weak classifier? 3) How to update the weight of the sample in online case? We will discuss the proposed solutions to these questions in the following.

(1)The number of training times for each sample to update the weak classifier

We use the idea in reference [12], where the sample is modeled by a Poisson distribution and each is used N times to update the weak classifier, where N is a random number generated by Poisson (ω) and ω is weight of the sample.

(2)Choosing weak classifiers in online mode

For the purpose of tracking, different weak classifiers (features) should be chosen to represent the appearance variations. In order to solve this problem, we introduce base classifiers which can be considered as a set of weak classifiers. The main idea is to apply online semi-supervised boosting not directly to the weak classifiers but to the base classifiers.

Given the base classifier $B_t = \{ f_{t,n}(x) \mid n = 1,2,\cdots,M \}$, $t = 1,2\cdots T$ .Where, $f_{t,n}(x)$ is the weak classifier. When a new sample comes, each weak classifier in the base classifier is updated and the best one is chosen to construct the strong classifier.

In order to choose the best one, we should calculate the classification error rate for all the samples. However, we can not obtain all samples in online mode. So we turn to calculate the classification error rate for the samples that have been come so far. If we use $S_{t,n}^c$ and $S_{t,n}^w$ to record the weights sum of the correctly and wrongly classified samples that the weak classifier $f_{t,n}(x)$ has been seen so far respectively, the error rate for the weak classifier $f_{t,n}(x)$ will be: $e_{t,n} = \dfrac{S_{t,n}^w}{S_{t,n}^c + S_{t,n}^w}$ .Thus, the best weak classifier in base classifier B_t is $f_t(x) = \arg\min_n \{e_{t,n}\}$.

(3) Updating the weight of the sample

If we have chosen the weak classifier $f_t(x)$ in B_t ,the weight of the current training sample should be updated before it is used to choose the next weak classifier in B_{t+1} .In this paper, we use the classification result of the $f_t(x)$ to update the weight of the sample. If $f_t(x)$ gives a correct classification, the weight is reduced. Else, it will be increased.The weight is updated as:

$$\omega = e^{-y[F(x)+f_t(x)]} \tag{10}$$

where, $F(x)$ is the strong classifier obtained after $t-1$ iterations ($t-1<T$), y is the label or pseudo label for the samples.

Finally, our online semi-supervised algorithm is as following:

Input: Training sample， $< x, y >, x \in X, X = X_L \cup X_U$

Initialization: The strong classifier $F(x) = 0$ ， $S_{t,n}^c = S_{t,n}^w = 0$;

Step 1 Obtain the weight and the label/pseudo label of the sample:

If $x \in X_L$, then the weight is $\omega_t = e^{-yF(x)}$ and the label is $y_t = y$;

If $x \in X_U$,then the pseudo label is $y_t = sign[\tilde{y}\cosh(F(x)) - \sinh(F(x))]$, the weight is $\omega_t = \lambda \mid \tilde{y}\cosh(F(x)) - \sinh(F(x)) \mid e^{-\tilde{y}F(x)}$

Step 2 Calculate the number of training times of the sample $< x, y >$

Step 3 Choose the best weak classifier from the base classifier B_t :

(a) Update each weak classifier $f_{t,n}$ (any method can be used), $n = 1,2,\cdots,M$;

(b) Classifier the sample x :If $f_{t,n}(x) = y_t$ ， then $S_{t,n}^c = S_{t,n}^c + \omega_t$; else $S_{t,n}^w = S_{t,n}^w + \omega_t$

(c) Choose $f_t(x) = \arg\min_{f_{t,n}} \{\dfrac{S_{t,n}^w}{S_{t,n}^c + S_{t,n}^w}\}$

Step 4 Update the weight of x : $\omega_t = e^{-y_t[F(x)+f_t(x)]}$

Step 5 $F(x) \leftarrow F(x) + f_t(x)$

Step 6 Repeat Step2~Step5 for all base classifiers $\{B_t \mid t = 1, 2 \cdots, T\}$

Output: $F(x) = \sum_{t=1}^{T} f_t(x)$

3. Visual Tracking based on Semi-supervised Boosting

Instead of considering tracking as an one-shot semi-supervised learning problem (as in reference [9]), which makes the tracker too tight to adaptive enough, we propose a novel tracking framework that treats samples differently when updating the classifier under different conditions. Specifically, when the classifier is sure that it has detected the object, samples extracted are considered as labeled. Otherwise, the samples are treated as unlabeled and are given pseudo labels. Figure 1 shows the framework.

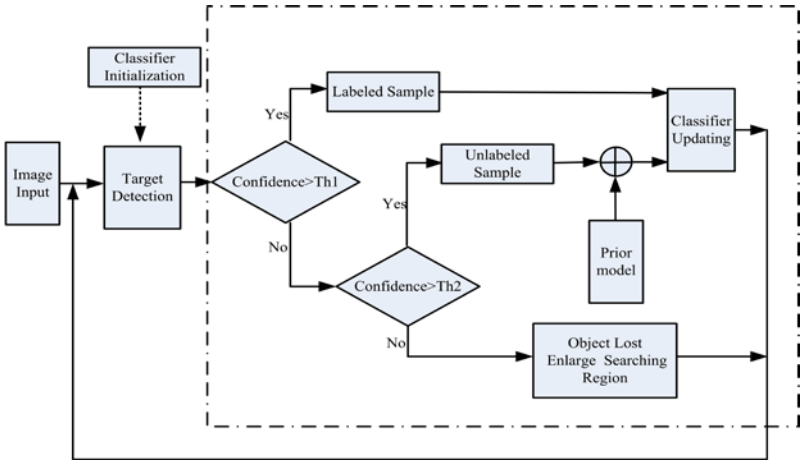


Figure 1.The Framework of Visual Tracking based on Online Semi-supervised Boosting

3.1. Prior Model and Classifier Initialization

Prior Model, namely the prior possibility distribution $\tilde{P}(y \mid x)$, which is a guider when deciding the pseudo labels of unlabeled samples. Any classifier trained by labeled samples can be used as the prior model. In this paper we consider prior model construction and classifier initialization as the same process. Suppose the target has been detected in the first frame, then two same boosting classifiers can be trained by the positive and negative samples extracted in the first frame. The two classifiers are considered as prior model and target detector respectively. The difference is that there will be no change for the prior model in subsequent frames while the target detector will be always updated according to its detection results.

3.2. Process of Target Tracking

The process of target tracking contains two steps: target detection and classifier updating.

(1)Target detection

When a new frame inputs, the detector searches and evaluates exhaustively in the region of interest and obtains a confidence map. The map is then filtered by a Gaussian filter and the corresponding location of the maximum value is considered as the position of the target.

(2) Classifier updating

Once the target has been detected, the classifier has to be updated. In our tracking framework, samples are treated differently when updating the classifier under different conditions. Specifically, two thresholds are used on the samples. If the confidence value of the detected target is greater than $Th1$, then samples extracted for updating the classifier are considered as labeled samples. If the confidence value of the detected target is less than $Th1$ but great than $Th2$, then all the samples are considered as unlabeled and are given pseudo labels. Else if the confidence value is less than $Th2$, then the target is lost and search region is enlarged in the next frame. Figure 2 shows the sample extracting process.

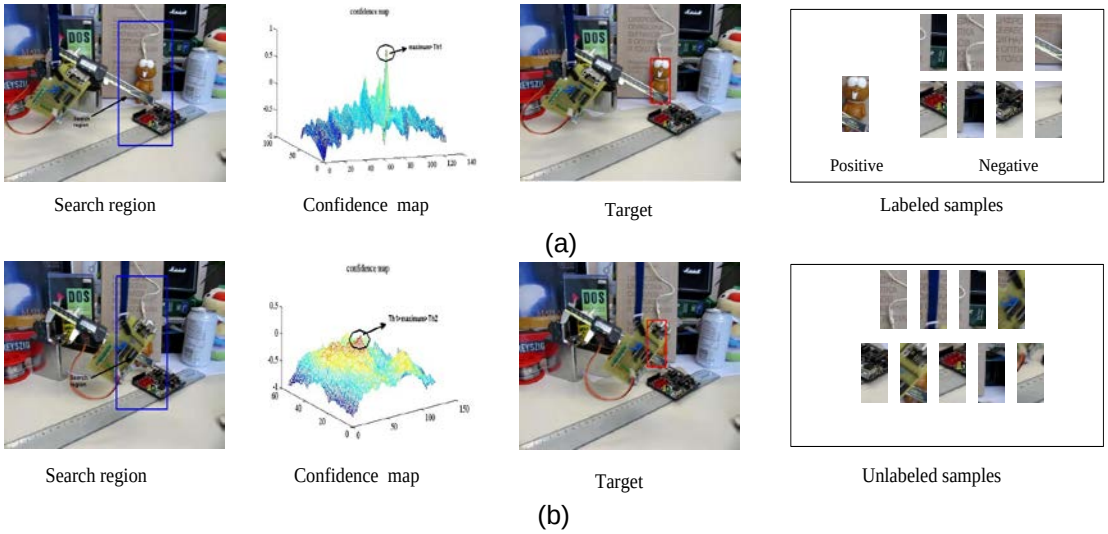


Figure 2. Sample Extracting: (a) Samples Extracting when Confidence Value of the Target is Greater than $Th1$, (b) Samples Extracting when Confidence Value of the Target is Less than $Th1$ but Greater than $Th2$

(3) Implementation Details

We use 100 base classifiers and each contains 150 harrlike weak classifiers (any other weak classifiers can be used). Let h_k is the k-th Harrlike feature, then the corresponding weak classifier is:

$$f_k = sign(P(y=1|h_k(x)) - P(y=-1|h_k(x))) \tag{11}$$

Where, $P(h_k(x)|y=1) \square N(\mu_1, \sigma_1)$, $P(h_k(x)|y=-1) \square N(\mu_{-1}, \sigma_{-1})$. Let $P(y=1) = P(y=-1)$ and Bayes rule is used to compute f_k .Initialize $\mu_1 = \mu_{-1} = 0$, $\sigma_1 = \sigma_{-1} = 0$,and when the classifier receives a positive sample, the following updating rule is used. The updating rule for μ_{-1} and σ_{-1} is as same.

$$\mu_1 \leftarrow \gamma h_k(x) + (1-\gamma)\mu_1$$

$$\sigma_1^2 \leftarrow \gamma(h_k(x) - \mu_1)^2 + (1-\gamma)\sigma_1^2$$

Where, γ is the updating parameter and can be obtained by Kalman filtering.

4. Experimental Results and Discussion

We test our method from four different aspects. Firstly, we investigate the parameter sensitivity of parameter λ .Then, the anti-drifting and adaptivity ability are tested separately. Finally, we investigate whether our track can significantly alleviate the drifting problem and keep adaptive enough to appearance variations simultaneously. We also compare it with three other state-of-the-art trackers.

4.1. Parameter Sensitivity

There is only one parameter λ in our online semi-supervised boosting algorithm, which is the contribution parameter of the unlabeled data. In this experiment, three videos with different length are used and the number of labeled data is the same. The tracking accuracy is calculated when λ varies from 1 to 10. From the results shown in Figure 3, we can see that the fluctuating range of the accuracy is very small for all the three videos, which is only 0.02. Thus, our online semi-supervised boosting algorithm is not sensitive to parameter changing.

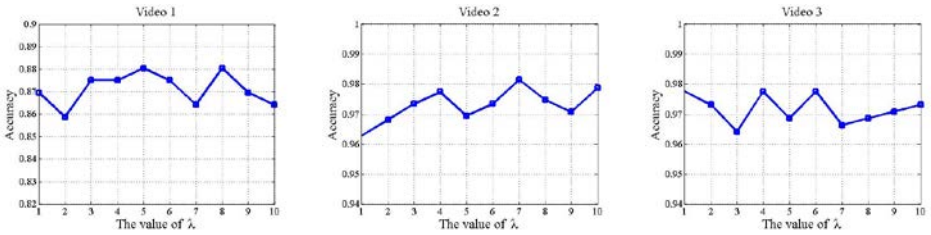


Figure 3.Tracking Accuracy with Different Value of λ

4.2. Anti-drifting Ability

In order to test the anti-drifting ability separately, we use the video in reference [14] .The target is a non-moving doll. Except a moving circuit board occludes the doll occasionally (6 times), nothing is changing. Figure 4 shows the tracking errors (in pixel) after several occlusions and Figure 5 demonstrates tracking results in the first occlusion. Although there are many occlusions happening, our tracker can still track the doll accurately. The mean tracking error is only 0.23. So when considering anti-drifting separately, our method can significantly alleviate drifting.

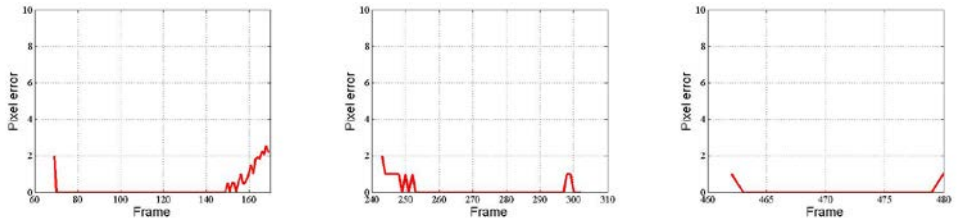
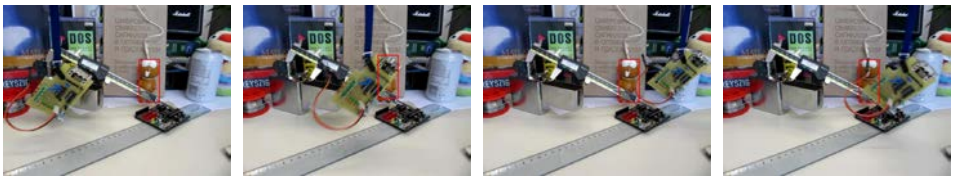


Figure 4. Tracking Errors after Occlusions: (a) After First Occlusion, (b) After Second Occlusion, (c) After Fifth Occlusion



(a) (b) (c) (d)

Figure 5. Tracking Results in First Occlusion: (a) Frame 2, (b) Frame 35, (c) Frame 86, (d) Frame 160

4.3. Adaptivity

For testing adaptivity separately, four videos with large appearance variations are chosen. Their main challenges are shown in Table 1. We also count the accuracy rate, losing rate and false positive rate of the four videos. In order to obtain accurate statistics, we define the score in formula (12).Where, ROI_D is the detected bounding box, ROI_G is the ground truth bounding box. When this score exceeds 0.5, we interpret the frame as true positive. Table 2 is the statistics. The results show our method has high accuracy, low losing and false positive rate when the target has large appearance variations. The average accuracy of our method on the four videos is 97.52% and the average losing rate is only 2.48%. Figure 6 shows the tracking errors. Figure 7 is the tracking results. From row 1 to row 4 are ThreePastshop1cor, Box, Woman and Sylvester respectively. So when considering adaptivity separately, our method is adaptive enough to appearance variations.

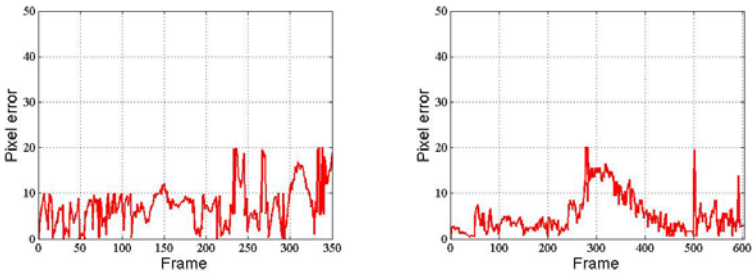
$$Score = \frac{area(ROI_D \cap ROI_G)}{area(ROI_D \cup ROI_G)} \tag{12}$$

Table 1. Videos and Challenge

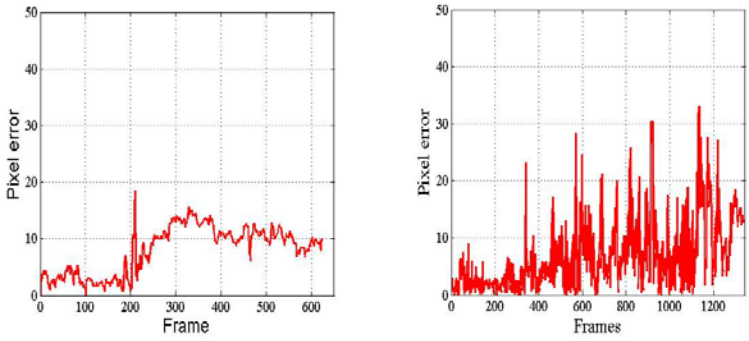
Video	Frame	Main challenge
Box[14]	350	3D motion, Motion blur
ThreePast Shop1cor[15]	604	Pose change, Illumination change
Woman	624	Appearance change, Pose change
Sylvester[14]	1344	3D motion, Illumination change

Table 2. Performance Statistics

Video	Accuracy rate	losing rate	False positive rate
Box	0.963	0.037	0
ThreePast Shop1cor	0.989	0.011	0
Woman	0.971	0.029	0
Sylvester	0.978	0.022	0



(a) (b)



(c) (d)

Figure 6. Tracking Errors of the Four Videos with Appearance Variations: (a) Box, (b) ThreePastShop1cor, (c) Woman, (d) Sylvester

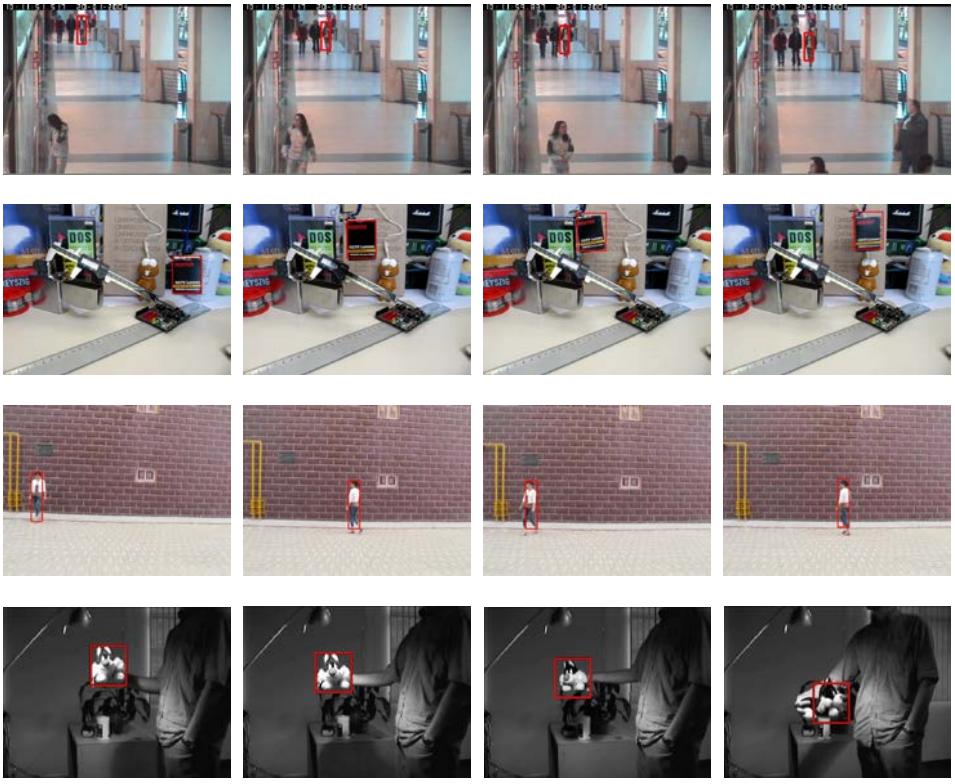


Figure 7.Tracking Results of the Four Videos with Appearance Variations: From Row 1 to Row 4 are ThreePastshop1cor, Box, Woman and Sylvester Respectively

4.4. Comprehensive Performance Compariation with other Trackers

This experiment investigates whether our track can significantly alleviate the drifting problem and keep adaptive enough to appearance variations simultaneously. We also compare our tracker with other three state-of-the-art trackers: Adaboost [7], MIL [13] and Semiboost [9]. Four videos under different circumstances are used (Table 3). In these videos, there are occlusion, fast appearance variations, similar object, *etc*.

Table 3. Four Videos and their Main Challenge

Video	Face[16]	OneLeaveShopReenter1cor[15]	Boy	WalkByShop1cor[15]
Frame	671	222	178	990
Main challenge	Moving camera, Occlusions	Pose change, Illumination change, Occlusions	Similar object, Illumination change, Occlusions	Pose change, Illumination change, Occlusions

Table 4. Losing Rate Comparison for Semiboost and Proposed Method

Video	Face	OneLeaveShopReenter1cor	Boy	WalkByShop1cor
Semiboost	11.62%	12.61%	41.57%	44.65%
Proposed	1.79%	0%	0.56%	1.72%

(1)Face Sequence

In this sequence the face of the woman is occluded occasionally by a book. Figure 8 and Figure 9 show the tracking results compariation of the four trackers. After the first occlusion, the Adaboost and MIL trackers are drifting away. Their average tracking errors are 23.69 and 15.22 respectively .The average tracking error of Semiboost tracker is 6.23, but its losing rate is 11.62%(Table 4). The tracking error of our tracker is 4.67 and the losing rate is only 1.79%. We can see that our tracker provides best performance than other three trackers.

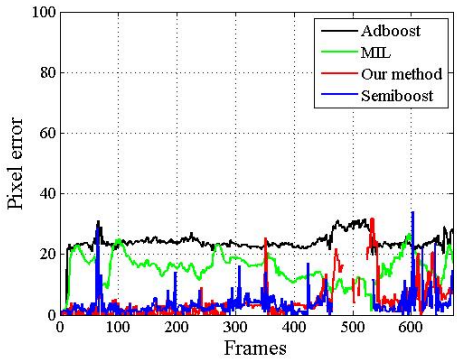


Figure 8. Tracking Errors of Face Sequence

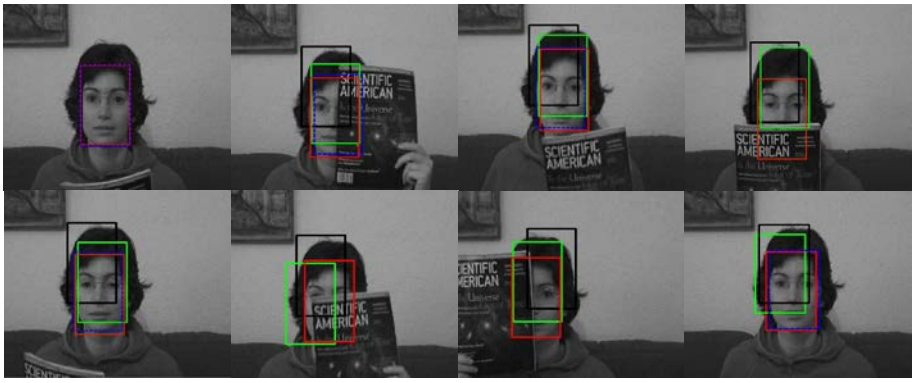


Figure 9. Tracking Results of Face Sequence: Our tracker (red), MIL (green), Adaboost (black), Semiboost (dotted blue)

(2)OneLeaveShopReenter1cor

There are illumination change, pose variations and a short occlusion in this sequence. Because of the appearance variations, the Semiboost tracker loses the target for 28 frames and its losing rate is 12.61% (Table 4).Although the occlusion is short, it increases the noises when updating the MIL and Adaboost trackers. As the noises accumulating, they drift away from the target. The average tracking errors of the two trackers are 8.29 and 5.82. Our tracker still does best in this sequence. Its average tracking error is 3.23 and it doesn't lose the target in any frame. Figure 10 and Figure 11 demonstrate the compariation.

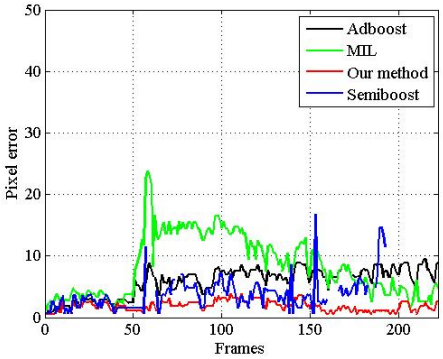


Figure 10. Tracking Errors of OneLeaveShopReenter1cor

Figure11. Tracking Results of OneLeaveShopReenter1cor: Our tracker (red), MIL (green), Adaboost (black), Semiboost (dotted blue)

(3)Boy sequence

The Boy sequence is our own. Except occlusion and fast appearance variations, there are two similar objects (a boy and a girl) in the sequence, and the boy is our target. As can be seen in Figure 12, MIL and Adaboost trackers begin to track the girl and drift far away after the girl occludes the boy in frame 78. Although the Seimboost tracker doesn't drift, its losing rate is too high (41.57% in Table 4) and it can't adapt to appearance variations. To the contrary, our tracker only loses the target once and tracks the target accurately and robustly. So our tracker not only can handle appearance variations but also has strong anti-drifting ability. Figure 13 shows the tracking results in some frames.

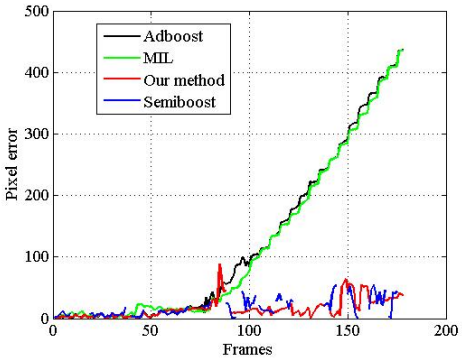


Figure 12. Tracking Errors of Boy Sequence

Figure 13. Tracking Results of Boy Sequence: Our Tracker (red), MIL (green), Adaboost (black), Semiboost (dotted blue)

(4) WalkByShop1cor

The main challenges of this sequence are illumination variation, pose change and occlusion. The results in Figure 14 and Figure 15 show our method has superior performance in handing drifting as well as appearance variations. Although the tacking error in some frames of Semiboost is similar to our method, its losing rate is 44.65%. The losing rate of our tracker is only 1.72 % (Table 4). The average tracking errors of MIL and Adaboost are 32.83 and 30.27, but the average tracking error of our method is only 10.51.

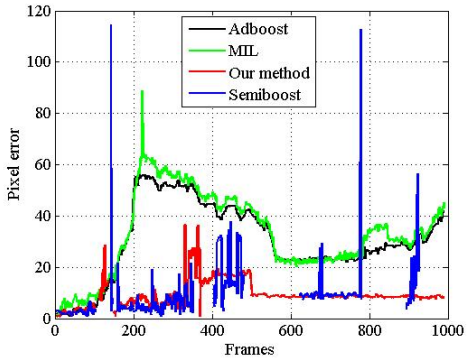


Figure 14. Tracking Errors of WalkByShop1cor

Figure 15. Tracking Results of WalkByShop1cor: Our Tracker(red),MIL(green), Adaboost(black), Semiboost (dotted blue)

5. Conclusions

In this paper, we address the anti-drifting and adaptivity of robust tracking. Based on a novel online semi-supervised boosting method, we propose a tracking framework that treats samples differently when updating the classifier under different conditions. In the experimental part, we compare our method with state-of-the-art tracking methods on challenging videos. Results demonstrate that our tracker can significantly alleviate the drifting problem and keep adaptive enough to appearance variations.

Acknowledgement

This work is supported by National Natural Science Foundation of China (61273277), Shandong Provincial Natural Science Foundation (ZR2011FM032), and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (20101174).

References

[1] A. Bakhtari, M. Mackay and B. Benhabib, "Active-vision for the autonomous surveillance of dynamic, multi-object environments", Journal of Intelligent Robot System, vol. 54, no. 4, **(2009)**, pp. 567-593.
[2] P. Vadakkepat, P. Lim and L. Desilva, "Multimodal approach to human-face detection and tracking", IEEE Transactions on Industrial Electronics, vol. 55, no. 3, **(2008)**, pp. 1385-1392.
[3] B. Matei, H. Sawhney and S. Samarasekera, "Vehicle tracking across non-overlapping cameras using joint kinematic and appearance features", The 24th IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, USA, **(2011)** June 21-23.
[4] Z. Kalal, K. Mikolajczyk and J .Matas, "Tracking-learning-detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, **(2012)**, pp. 1409-1422.
[5] H. X. Yang, L. Shao, F. Zheng, L. Wang and Z. Song, "Recent advances and trends in visual tracking: a review", Neurocomputing, vol. 74, no. 18, **(2012)**, pp. 3823-3831.
[6] M. Tian, W. Zhang and F. Liu, "On-line ensemble SVM for robust object tracking", 8th Asian Conference on Computer Vision, Tokyo, Japan, **(2007)** November 18-22.
[7] H. Grabner and H. Bischof, "On-line boosting and vision", Computer Vision and Pattern Recognition, New York, USA, **(2006)** June 17-22.
[8] A. Saffari, C. Leistner, J. Santner, M. Godec and H. Bischof, "On-line random forests", Computer Vision Workshops, Tokyo, Japan, **(2009)** September 27-October 4.
[9] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking", European Conference on Computer Vision, (2008) June 24-26; Alaska, USA.
[10] A.Saffari,H.Grabner and H.Bischof, "SERBoost:semi-supervised boosting with expectation regularization", European Conference on Computer Vision,(2008) October 12-18; Marseille, France.
[11] J.Friedman,T.Hastie,and R.Tibshirani, "Additive logistic regression: a statistical view of boosting", The Annals of Statistics,vol.28,no.2,(2000),pp.337-407.

[12] N. Oza and S. Russell, "Online bagging and boosting", 2005 IEEE International Conference on Systems, Man and Cybernetics, Hawaii, USA, **(2005)** October 10-12.
[13] B. Babenko, M. Yang and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, **(2011)**, pp. 1619-1632.
[14] J. Santner, C. Leistner, A. Saffari, T. Pock and H. Bischof, "PROST: Parallel Robust Online Simple Tracking", Computer Vision and Pattern Recognition, San Francisco, USA, **(2010)** June 13-18.
[15] CAVIAR Database, http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA2/.
[16] A. Adam, E. Rivlin and I. Shimshoni, "Robust fragments based tracking using the integral histogram", Computer Vision and Pattern Recognition, New York, USA, **(2006)** June 17-22.

Authors

Wenhui Dong received her B.S. degree in electronic engineering from Qufu Normal University in 2003 and M.S. degree in communication and information systems from Shandong University in 2006.Now she is a PhD candidate student in school of control science and engineering, Shandong University. She focuses on computer vision, pattern recognition and image processing.

Faliang Chang received his B.S. degree, M.S. degree and Ph.D degree in Automation from Shandong University in 1986, 1989 and 2005 respectively. Now he is a professor and Ph.D. candidate tutor in school of control science and engineering, Shandong University. His research interests are computer vision, pattern recognition and image processing.

Zijian Zhao received the B.S.degree and M.S. degree in electric engineering from Shandong University, in 2002 and 2003 respectively, and the Ph.D. degree in image processing and pattern recognition from Shanghai Jiao Tong University, in 2009. He was a research scientist at University of Oulu in 2010. He was a Research Engineer at TIMC-IMAG, University Joseph Fourier from 2010 to 2012. In Mar. 2012, he joined Shandong University as a Docent. He focuses on computer vision, artificial intelligence and pattern recognition.