One-Class SVM Assisted Accurate Tracking

One-Class SVM Assisted Accurate Tracking

Keren Fu*, Chen Gong*, Yu Qiao*, Jie Yang*, Irene Gu†
*Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University,
and Key Laboratory of System Control and Information Processing,
Ministry of Education of China, Shanghai 200240
†Signal Processing Group, Department of Signals and Systems,
Chalmers University of Technology, Gothenburg, 41296, Sweden

***Abstract*—Recently, tracking is regarded as a binary classification problem by discriminative tracking methods. However, such binary classification may not fully handle the outliers, which may cause drifting. In this paper, we argue that tracking may be regarded as one-class problem, which avoids gathering limited negative samples for background description. Inspired by the fact the positive feature space generated by One-Class SVM is bounded by a closed sphere, we propose a novel tracking method utilizing One-Class SVMs that adopt HOG and 2bit-BP as features, called One-Class SVM Tracker (OCST). Simultaneously an efficient initialization and online updating scheme is also proposed. Extensive experimental results prove that OCST outperforms some state-of-the-art discriminative tracking methods on providing accurate tracking and alleviating serious drifting.**

I. Introduction

Tracking is regarded as a key point in computer vision field and has been extensively researched for decades. Recently, tracking-by-detection methods [1-9] are explored to formulate tracking as a binary classification problem, which distinguishes the object from the background. That is, the target regions are regarded as positive samples and the non-target regions are deemed as negative samples while a classifier is trained to seek a decision boundary that can best separate the positive and the negative. The classifiers used to tackle this problem, like SVM [2] or the ones generated by Adaboost algorithm [3], usually have good ability to handle high-dimensional data. Babenko et al. [6] adapt Multiple Instance Learning (MIL) [12, 13] instead of traditional supervised learning by building an evolving boosting classifier that tracks bags of image patches, and report excellent tracking results on challenging video sequences. However, these supervised or MIL based methods usually generate an open positive feature space and sometimes may be not robust to the outliers [10].

The semi-supervised learning based methods [7-9] are then proposed to treat object tracking as an online semi-supervised binary classification problem. In addition to the labeled samples, the semi-supervised classification tries to use more unlabeled samples, which brings stronger ability to handle the outliers. The deficiency of these methods is that beyond the labeled data, large number of unlabeled data should be collected online, and also many semi-supervised algorithms like transductive support vector machine [10] are highly computational, hence degrading the performance of the tracking system to be far away from real-time processing. Besides, the semi-supervised based tracking methods have not totally solved the outlier problem as well, which may lead to drifting problem. This will be discussed in the following section.

In this paper, we propose an accurate tracking method using One-Class SVM, so we call it the *One-Class SVM Tracker* (OCST), which falls into the tracking-by-detection category. We propose that the tracking problem may be treated as one-class classification case rather than the binary case. One-Class SVM [14] is proposed to estimate the distribution of high-dimensional data and then it has been used in document classification [15] and image retrieval [16]. Recently, [17] uses two competing One-Class SVMs to segment foreground from video sequences. The most related work to ours is [25] which employs One-Class SVM in visual object tracking. However One-Class SVM in [25] is more like a refiner rather than classifier. It is used to determine the target's location using the information from samples which are similar to the target in the last frame. So their method may not be regarded as discriminative method. In contrast, we introduce One-Class SVM as discriminative classifier while take advantage of its ability of dealing with outliers and processing high-dimensional data. Using One-Class SVM for tracking has three-fold advantages:

More accurate position estimation: Only the real target region will be classified as inlier or characterized with highest classification score (see Fig.1), thus locating the target region accurately and enormously pulling down the false positive rate.

Alleviate the drifting problem of binary classification: Supervised or semi-supervised methods need to update the training samples or the decision boundary online. If samples with large locating deviation are added into the positive training sample set or used to update decision boundary, the drifting problem will come. Thus accurate position estimation provided by OCST may alleviate such drifting problem.

Lower computational cost: The training samples of One-class SVM are only the positive. So no more negative or unlabeled samples are needed, which reduces the burden of the classifier as well as computational cost.

The rest of this paper is organized as follows. Related work is described in section 2. Details of our implementation with One-class SVM are demonstrated in section 3. Experimental results are analyzed in section 4 while conclusion and future work are drawn in section 5.

II. Related Work

A. Tracking-by-detection Methods

Tracking-by-detection methods, or so-called discriminative methods are explored to formulate tracking task as a binary classification problem and supervised or semi-supervised learning methods are considered. The support vector tracker [2] (denoted as SVT afterwards) uses an offline-learned SVM as classifier and embeds it into an optical flow to track moving vehicles. The final tracking position is characterized with the highest SVM score. Yet their SVM never updates online, leading to lower adaptability. In addition, the effort of building such large off-line sample set manually is usually considerable. [3] utilizes the Adaboost algorithm to perform online feature selection. Their positive and negative samples are collected online, so no additional manual effort is needed. However, their method may cause drifting problem in complex background, due to potential effect of the outliers [10], which is also the common problem of these supervised methods. [6] uses Multiple Instance Learning (MIL) [12,13] to train the appearance classifier, resulting in relatively robust tracking and an online boosting algorithm for MIL is also presented. Then [8] adopts co-training to take the advantage of multiple independent features for training a set of classifiers online. The classifiers then collaboratively classify the unlabeled data and use this newly labeled data to update each other. Each feature is used to train an online SVM, and their outputs are combined to give the final classification results. [9] uses off-line detector, on-line supervised identifier and semi-supervised tracker to extend semi-supervised tracking by object specific and adaptive priors. However, their model relies strongly on the prior classifier, leading to frequent target loss.

These tracking-by-detection methods mentioned above treat tracking as binary classification problem, no matter supervised learning based, MIL based or semi-supervised learning based. So their common problem is that they can not fully handle the outliers (Fig.1), leading to inaccurate tracking. Also, the semi-supervised learning requires a large number of unlabeled samples for learning, simultaneously with extra time cost in feature extraction and classification.

B. Our Motivation

The key insight of our approach is to take advantage of One-Class SVM for tackling tracking problem. A vivid illustration of the difference between One-Class SVM and supervised binary classifier as well as semi-supervised binary classifier is shown in Fig.1. In Fig.1, the circles represent the positive samples and the crosses stand for the negative ones. The solid balls represent unlabeled samples while the diamond stands for a special unlabeled sample. We can see that no matter using supervised or semi-supervised learning method, when training samples are collected, a decision boundary may be generated to classify the positive and negative categories with the maximum margin or the minimum error. However, an unlabeled sample (the diamond one) which is far away from the positive sample set may be classified as positive (Fig.1 (a) and (b)). This unlabeled sample may correspond to the region which contains a non-target object or part of the tracked target. Moreover, high classification score may be obtained for this unlabeled sample due to its further distance from the decision boundary, thus it is more likely to contribute much to the final tracking position, leading to inaccurate location or drifting.

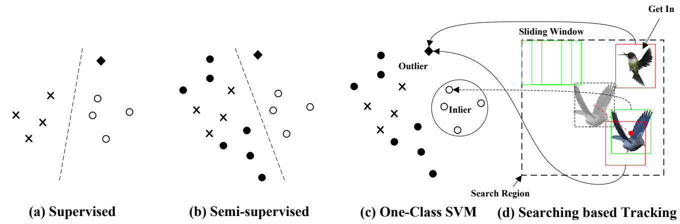

Fig. 1. An illustration of the difference between using One-Class SVM and supervise binary classifier as well as semi-supervised binary classifier. The circles represent the training positive samples and the crosses stand for the negative ones. The solid balls represent unlabeled samples and the diamond stands for a special unlabeled sample. As the feature space of the target generated by One-Class SVM is bounded by a closed sphere, only the real region of the tracked target will be classified as inlier by One-Class SVM.

The key point of the problem above is that the feature space for the target separated by the decision boundary in binary case is usually an *open interval* (Fig.1 (a) and (b)). Moreover, using limited number of labeled negative or unlabeled samples can hardly describe or estimate the total complex moving environment. So One-Class SVM should be introduced to solve this problem. It can estimate the distribution of high-dimensional samples and the classifier only needs positive samples as input and hence effectively avoids collecting negative samples for background description. From another sight, One-Class SVM makes the judgment whether an unlabeled sample is the target object or the remaining, using the only information of positive training samples.

Another important point which is declaimed in this paper is that the feature space of the target should be bounded by a *closed sphere*. Thus an unlabeled sample is classified as "inlier" or "outlier" rather than "positive" or "negative". In Fig.1, all samples far away from the positive sample set or the hyper sphere should be classified as outliers. Only the real region of the tracked target will be classified as inlier (Fig.1 (c) and (d)), which results in more accurate tracking as well as alleviating the drifting risk.

III. One-Class SVM Assisted Tracking

In this section we introduce our tracking algorithm, the OCST, which uses One-Class SVM as discriminative classifier and takes the advantage of the dense HOG feature [11] and 2bitBP feature [24]. We begin with a brief description of One-Class SVM. Next we illustrate the details of feature extraction and combination. Finally, we review our online tracking framework.

A. One-Class SVM

The SVM algorithm as it is usually construed is essentially a binary-class algorithm [18] (needs negative and positive samples). However, when only the positive samples can be acquired while the negative samples have no certain distribution and remain irregular, One-Class SVM should be considered. One-Class SVM algorithm [14] maps the data into a feature space H using an appropriate kernel function, and then trying to separate the mapped vectors from the origin with maximum margin.

Here, let $x_1, x_2, x_3...x_l$ be training samples belonging to one known class X, where X is a compact subset of R^N [14]. Let $\phi : X \to H$ be a kernel map which transforms the training samples to another space. Then, to separate the data set from the origin, one needs to solve the following quadratic programming problem:

$$\min \ \frac{1}{2}||w||^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - \rho \tag{1}$$

$$\texttt{s.t.} \ w\phi(x_i) \geq \rho - \xi_i, \ i = 1, 2, 3, ..., l, \ \xi_i \geq 0 \tag{2}$$

Nonzero slack variables ξ_i are penalized in the objective function. The decision function corresponding to w and ρ is

$$f(x) = w\phi(x) - \rho \tag{3}$$

Equation (3) will be positive for most samples x_i contained in the training set. $v \in (0, 1)$ is a parameter which controls the number of samples contained in the hyper sphere.

Based on this theory, in tracking task, we also search for a hyper sphere which contains most of the training samples obtained consequently from the target region. After training, the decision boundary may allow us to choose the most appropriate candidate region.

B. Feature Selection

Beyond the using of One-Class SVM, the feature selection for tracking is also an important part. Good features usually have nice ability to characterize the unique appearance of the tracked target meanwhile distinguish it from the complex background and other objects. In recent research, many kinds of features such as HOG [11], color histogram [19] , Haar-like feature [23] , Garbor feature [20] and LBP feature [21] are adopted for tracking. [8] uses both color histogram and HOG to train their corresponding SVMs while [3] uses the Haar-like, LBP and HOG. However, the color histogram has relatively weaker distinguishing ability, especially in the case that the background color is similar to the target color, thus leading to drifting, e.g. the Meanshift algorithm [19]. So in this paper, we reject the color histogram and tend to choose nice kind of features to describe the target's shape and texture, and the features selected should also be invariant to illumination changes.

We ultimately base our tracker on HOG and a new feature called 2bitBP [24] for characterizing the appearance of target. In this paper, we fuse these two feature extraction processes into the same scheme. First, the target region is divided into some overlapped square blocks, similar to the standard HOG

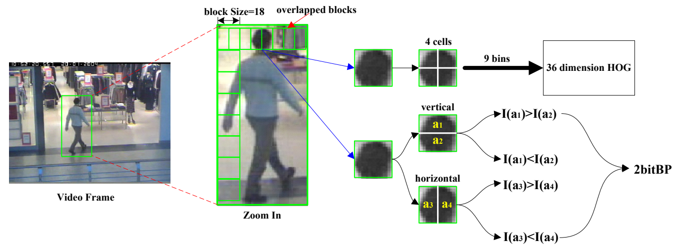

Fig. 2. An example of our feature extraction for a $W = 75$ and $H = 150$ target. The feature region is centered and forced to contain at least 4 non-overlapped blocks in the horizontal or vertical direction. In each block, a 36 dimension HOG and 2bitBP are generated.

extraction (Fig.2). In each block, we extract both HOG and 2bitBP. For HOG feature, each block contains 4 cells. If 9 bins are chosen in one cell, thus a 36 dimension vector is constructed in every block. For 2bitBP feature, the block is divided in horizontal and vertical direction, and the sum areas of the two sides are computed and compared to obtain a 2bit code.

In traditional HOG for human detection [11], the recommended block size is 16 for 64 by 128 human images. However, in tracking, the block size should be adaptive to the size of a specific tracked target. So we define the block size as

$$BS = \max\{\min(\frac{W}{N}, \frac{H}{N}), MinSize\} \tag{4}$$

where W and H are respectively the width and height of the tracking rectangle. We force the minimum number of block (non-overlapped) in the horizontal or vertical direction to be N. The default N is chosen as 4, which achieves the best performance in our experiment. $MinSize$ is the lower limit for the block size and is set to 8. A vivid example of our feature extraction for a $W = 75$ and $H = 150$ target is shown in Fig.2.

C. On-line Tracking

In the beginning of tracking process, an initial tracking rectangle should be given. This rectangle may be chosen manually or provided by an object detector. When the first rectangle is given, the tracker begins to work. As supervised and semi-supervised tracking methods need to collect the positive and negative samples online in the first several frames. [8] adopts the meanshift tracker [19] to track target in the beginning several frames in order to collect positive and negative samples. Compared with their method, our method is much simpler but more effective.

We construct a sample pool to store the positive samples (each sample contains HOG and 2bitBP features). Then these samples are used to train our One-Class SVMs. By the way, we first build a score function which is similar with the SVM score in the binary classification case. Our score function is obtained using the decision function (3) as

$$S(x) = w_H e^{\alpha_H f_H(x)} + w_B e^{\alpha_B f_B(x)} \tag{5}$$

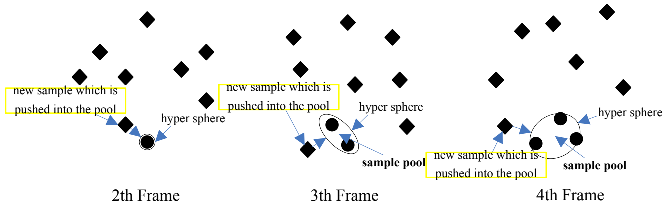

Fig. 3. The initializing process in the first several frames. In the 2nd frame, the unlabeled sample which is nearest to the decision sphere is chosen as our positive sample. The One-Class SVM may degrade into NN classifier. In the following frames, more and more positive samples which maximize (5) are pushed into the pool.

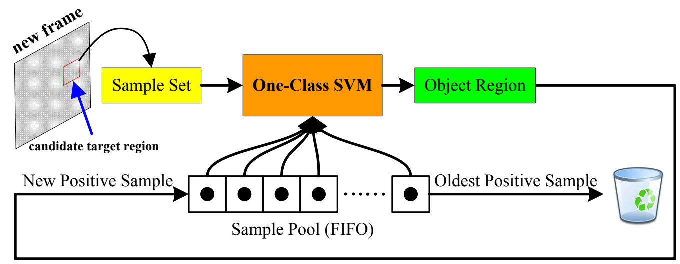

Fig. 4. Our on-line tracking algorithm based on One-Class SVM. The unlabeled sample set in new frame is extracted uniformly surrounding the last target position. The object region is chosen as the sample which maximizes (5). Our sample pool is realized using a FIFO. When the FIFO is full, adding new positive sample will cause old sample popped out.

where f_H and f_B are the One-Class SVM decision functions for HOG and 2bitBP respectively. α_H and α_B are scaling factors which help pull down (5) when the exponents of the two terms turn negative. In practice we find $\alpha_H = \alpha_B = 10$ to be suitable for normalized features. w_H and w_B are the weights computed using classification errors as

$$w_H = 1 - \epsilon_H/(\epsilon_H + \epsilon_B + \tau) \tag{6}$$

$$w_B = 1 - \epsilon_B/(\epsilon_H + \epsilon_B + \tau) \tag{7}$$

in which ϵ_H and ϵ_B are respectively classification errors of the two One-Class SVMs. τ is a small number which avoid the divide by 0 issue. $w_H + w_B = 1$ is also satisfied above.

From (5), we can see that $S(x) > 1$ roughly means the corresponding sample is classified as inlier while $S(x) < 1$ is for outlier. Since (5) helps us visualize the classification results, in practice, we always select the samples which make (5) achieve its maximum among unlabeled samples in each frame as our target and add them into the pool.

In the first frame, the sample pool is initialized (empty) and the first positive sample is pushed into the pool. The only sample is then used to train the classifiers. In this case, the One-Class SVM may degrade into a nearest neighbor classifier, which seeks a nearest neighbor in feature space of the only training sample. In the following frames, more and more samples which maximize (5) are collected online and pushed into the pool to train the One-Class SVMs.

The initializing process is shown in Fig.3. Gradually, the added target samples may increase the computational cost and burden of the classifiers. So we always remain the latest k (e.g. $k = 30$) samples in our sample pool and the relatively older samples are thrown away. Actually, we use a FIFO to realize this process. When the FIFO is full, adding new positive sample will cause old sample popped out. Fig.4 shows our on-line tracking algorithm based on One-Class SVM.

IV. Experiments and Analysis

Our method is validated on large amount of video sequences and compared with some state-of-the-art discriminative tracking methods including online boosting (OB) [3], beyond semi-boosting (BSB) [9] (the codes of these methods are available at the authors' webpage). In addition, a typical meanshift method [19] and a naive nearest neighbor tracker (NNT) are considered into comparison. Note in this paper, we use the typical motion model. That is we extract the unlabeled samples uniformly surrounding the detected target position in the last frame and use (5) to find the target in the current frame. This idea is also shown previously in Fig.1 (d) and Fig.4.

A. Tracking in Complex Background

Fig.5 shows the comparison results on sequence PETS2001, Railway and Parking lot. The challenge of tracking on these sequences are complex background and abrupt background changes. Meanshift drifts seriously when sudden background changes occur such as the pedestrian in PETS2001 gets out of the grassland and the buddy in Railway starts to cross the railroad.

Online boosting combines multiple features, so it is more robust to the complex background and environmental changes. However, it still treats tracking as a binary classification problem and thus, the tracking rectangle may sometimes drift, leading to inaccurate locating, as is shown in the 101st frame of Railway sequence and the 44th frame of parking lot sequence.

The beyond semi-boosting which combines off-line detector, supervised on-line identifier and semi-supervised tracker, is sometimes likely to be confused by target appearance changes. As the pedestrian in PETS2001 gets out of the grassland, the tracker makes the wrong decision that the target is lost, and starts to search for the target in frame 118 and then locates a wrong target in frame 119.

The NNT searches for the region which is nearest to the original target in feature space. However, the Euclidean distance may not really characterize the distance in the feature space, especially in high dimensional representation. So the inaccurate locating may happen.

Relatively, our method locates the target more accurately on these three sequences. Combining the dense HOG feature, 2bitBP feature and One-Class SVM classification, the shape and texture information are well extracted and only the real target region in the next frame will be found correctly, no matter the background changes abruptly or the sudden occlusion happens. The comparison between our OCST tracker and

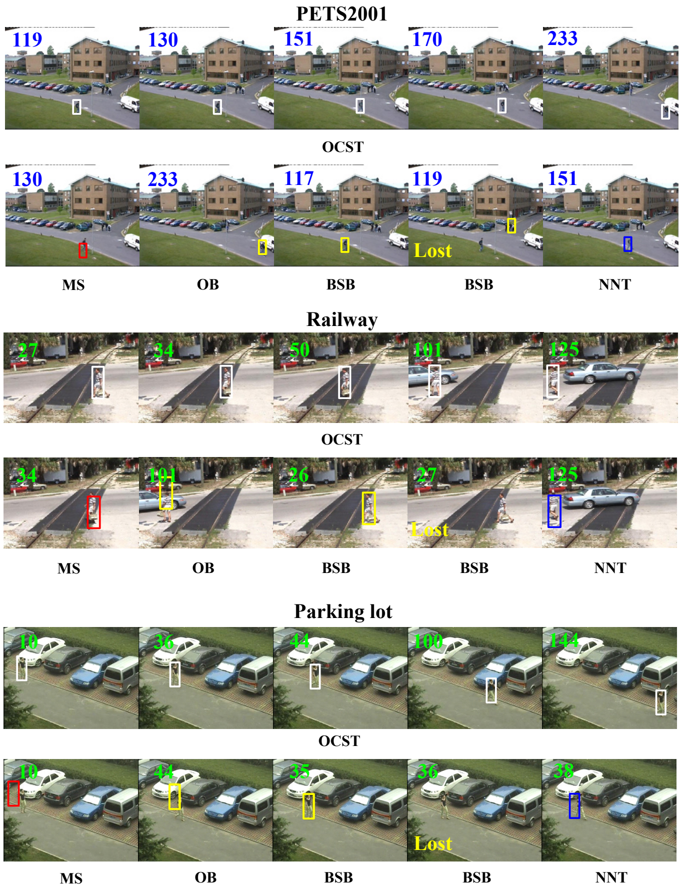

Fig. 5. Tracking performance comparison on sequence PETS2001, Railway and Parking lot.

another certain tracker can be obtained in the frame with the identical frame number in Fig.5.

B. Tracking Part of Human Body

Tracking certain part of the human like head, face or eyes is very important in identity recognition, video conference and user interaction. We track some part of human using the state-of-the-art methods as well as our OCST method. Usually, tracking certain part of human is tougher than tracking a single object which is rigid as we should consider the pose changes.

In the Girl sequence (Fig.6), we track the head of the girl. The challenges include the pan, tilt, zoom control, occlusion by another face, 360-degree rotation and the flesh-colored board in the background. Our method performs more satisfactory results even when the head leans and is partially occluded by some other face, while the tracking rectangles of other methods still drift when some obvious pose changes occur. As the ground-truth of this sequence is also available [26], we also measure the pixel-wise tracking errors of these methods. From the curves presented in Fig.6, it can be concluded that methods like MS, OB and NNT may occur serious drifting when the head leans and rotates while the BSB will lose target several times during the whole tracking process (values of the green curve in Fig.6 which are rendered 50 indicate the time when the BSB makes the wrong judgment that the target is lost). Compared with other methods, our approach provided steadier tracking with relatively lower error.

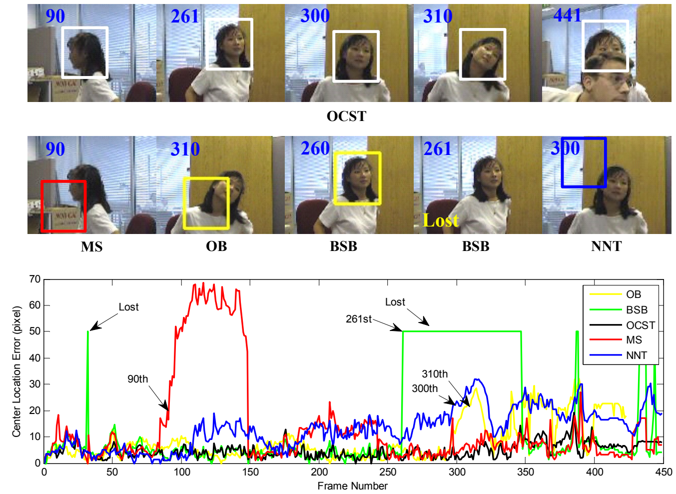

Fig. 6. Tracking performance comparison on Girl sequence. The tracking errors of different methods are also shown.

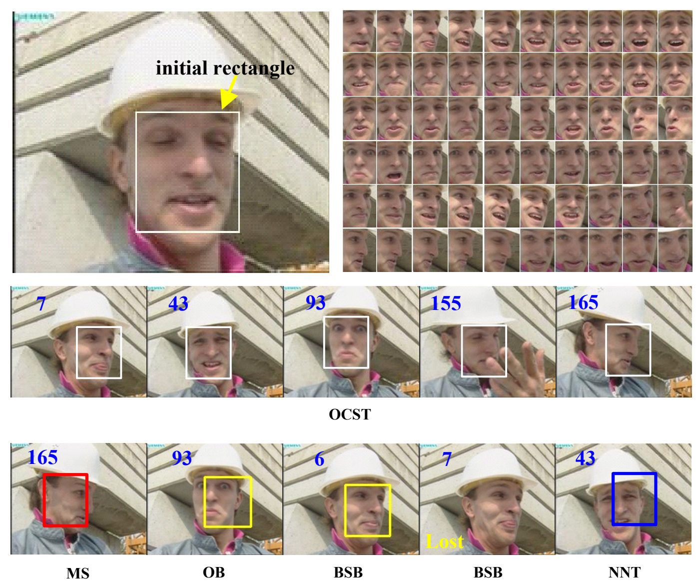

Fig. 7. Tracking performance comparison on Foreman sequence. The initial frame with tracking box is presented while the tracked face regions of our method extracted in every 3 frames are shown on the right.

In the Foreman sequence [27] (Fig.7), the face is tracked. Our method also provides impressive tracking result, no matter different head poses, exaggerated facial expression conversion and partial occlusion by waving hand occur. The initial frame with tracking box is presented while the tracked face regions of our method extracted in every 3 frames are shown on the right. Comparison between other state-of-the-art methods and our method on a certain frame also can be obtained in Fig.7.

In the Surfer sequence [28] (Fig.8), we also track the head part. As this sequence has a relatively monotonous background, the MS performs better than before, but as abrupt

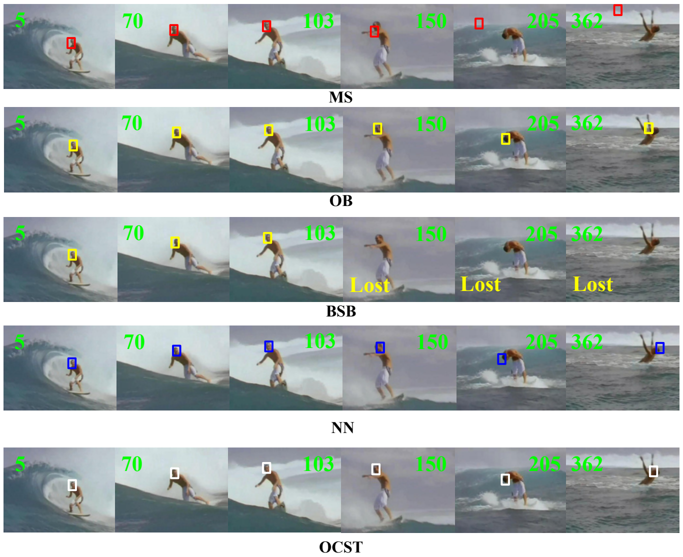

Fig. 8. Tracking performance comparison on Surfer sequence.

Table 1. Comparison in terms of the number of correctly tracked frames.

Sequence	Frames	MS	OB	BSB	NN	OCST
PETS2001	250	127	**250**	117	**250**	**250**
Railway	135	73	105	50	125	**135**
Parking lot	157	5	35	57	37	**157**
Girl	448	340	406	96	266	**448**
Foreman	180	**180**	179	75	110	**180**
Surfer	367	150	362	103	188	**367**
Average correct rate	N/A	54.1%	81.6%	35.2%	64.2%	**100%**

pose changes occur, the tracker drifts away. Both OB and OCST perform well on this sequence, noting that in the last frame OCST still sticks to the right position.

Finally, Table 1 shows quantitative results for all sequences. A frame is considered as correctly tracked if the real target rectangle overlap with the tracking rectangle is larger than 50%. Thus this criterion directly shows whether a tracker presents serious drifting during tracking process. For our method, slight deviation may be generated in some frames, but no serious drifting happens in the selected video sequences.

In summary, our method performs relatively accurate and stable tracking compared with other compared state-of-the-art methods. This should be attributed to the ability of One-Class SVM for dealing with high dimensional data. Besides, combining dense HOG feature and 2bitBP captures the fine shape and texture characteristics of the target. Regarding tracking as one-class classification problem may well handle the outliers and contribute to the alleviation of drifting problem to some extent.

V. Conclusion and Future Works

In this paper, we propose a novel tracking method using One-Class SVM. Combining the dense HOG feature and 2bitBP with One-Class SVM, OCST may well handle the outliers and alleviate drifting. A challenge for us in the future is trying to track articulated objects which cannot be easily delineated with a bounding box. These objects may require part-based appearance model, which may let us develop our OCST for part-based learning.

VI. Acknowledgement

We thank the anonymous reviewers for their valuable suggestions. This research is partly supported by National Science Foundation, China (No: 61273258).

References

[1] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L.V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in ICCV, 2009.
[2] S. Avidan, "Support vector tracking," PAMI, 26(8):1064-1072, 2004. 2
[3] H. Grabner and H. Bischof, "On-line boosting and vision," in CVPR, 2006.
[4] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in CVPR, 2009.
[5] S. Avidan, "Ensemble tracking," PAMI, 29(2):261-271, February 2007. 2, 6.
[6] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in CVPR, 2009.
[7] H. Grabner and H. Bischof, "Semi-supervised on-line boosting for robust tracking," In ECCV, 2008.
[8] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in ICCV, 2007.
[9] S. Stalder, H. Grabner, and L. Van Gool, "Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition," in Proceedings ICCV09 WS on On-line Learning for Computer Vision, 2009.
[10] X. Zhu and A. Goldberg, "*Introduction to Semi-Supervised Learning*," Morgan and Claypool Publishers, 2009.
[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005.
[12] P. Viola, J. Platt, C. Zhang, "Multiple instance boosting for object detection," In NIPS, 2005.
[13] T. G. Dietterich, R. H. Lathrop, and L. T. Perez, "Solving the multiple-instance problem with axis parallel rectangles," Artificial Intelligence, pp. 31-71, 1997.
[14] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the support of a high-dimensional distribution," Technical report, Microsoft Research, MSR-TR-99-87, 1999.
[15] L.M. Maneviitz, M. Yousef, "One-class SVMs for Document Classification," Machine Learning Research, Vol. 2, 139-154, 2002.
[16] Y. Chen, X.S. Zhou, And T.S. Huang, "One-class SVM for learning in image retrieval," in ICIP, 815-818, 2001.
[17] M.L. Gong, L. Cheng, "Foreground Segmentation of Live Videos using Locally Competing 1SVMs," in CVPR, 2011.
[18] V. Vapnik, "The Nature of Statistical Learning Theory," Springer Verlag, New York, 1995.
[19] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in CVPR, pages, 2000.
[20] T.S. Lee, "Image representing using 2D Gabor wavelets," PAMI, vol.18(10), pp. 959-971, 1996.
[21] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," PAMI, 24(7):971-987, 2002.
[22] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," PAMI, 2010.
[23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in CVPR, 2001.
[24] Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking." in OLCV, 2009.
[25] Li Li, Zhenjun Han, Qixiang Ye, and Jianbin Jiao, "Visual Object Tracking via One-Class SVM," in ACCV 2010 Workshop on Visual Surveillance.
[26] http://www.ces.clemson.edu/∼stb/research/headtracker/seq/
[27] http://media.xiph.org/video/derf
[28] http://vision.ucsd.edu/∼bbabenko/project_miltrack.shtml