An Object Tracking Method Using Extreme Learning Machine with Online Learning

Yuanlong Yu, Liyan Xie, and Zhiyong Huang
College of Mathematics and Computer Science
Fuzhou University
Fuzhou, Fujian, 350116, China
Emails: yu.yuanlong@fzu.edu.cn, 747516460@qq.com, hzy_fzusj@sina.com

***Abstract*—Target tracking is a challenging task in computer vision. It aims to detect and track particular objects in sequences. Illumination variation, motion of target, occlusion and background clutter make target tracking extremely challenging. We propose an novel online target tracking method which based on extreme learning machine(ELM). This tracking method consists of three modules: training, tracking and classifier update. The training stage aims to train ELM by using the training set. Extracting histograms of oriented gradients (HOG) features in the first frame of each sequence for training ELM. Then the tracking stage will make predictions about the object position and detect the target in candidate regions. A simple object motion model is designed to predict the object position. Finally, according to tracking results, the classifier can be updated for online learning. A large number of experimental results have validated this proposed method.**

I. Introduction

Target tracking is a significant topic in computer vision research field. It includes many applications like intelligent video surveillance, smart rooms, intelligent transportation, driver assistance and other fields. Target tracking can be divided into generative methods [1] [2] [3] [4] [5] and discriminative methods [6] [7] [8] [9].

Generative algorithms use some generative process to develop the target model and use it to find regions that are most similar to the object. The purpose of generative algorithms is to establish image representations to facilitate robust tracking. Some popular generative methods include the l_1 tracker [3], which represents the object by a sparse combination of overcomplete basis vectors and incremental visual tracking (IVT) [1]. And the IVT represents the target by learning an incremental subspace model. Since the background factor is not considered, generative methods are only suitable for less complex environments.

Compared with generative algorithms, discriminative trackers perform better in the case of background clutter and occlusion while considering the background as an important factor. These algorithms use a discriminative model to recognize the object from background and update the object model by new samples coming in. Some typical trackers in this category are tracking-learning-detection (TLD) [7] [10], online adaboost (OAB) [6] [11] [12] and multiple instance learning (MIL) [13]. Although widely used, these algorithms raise some issues. In term of updating, the performance of the classifier greatly depend on the updating samples. Some trackers [6] [14] [15] only use some negative samples and positive samples to update the model. As the target model updates with noisy and potentially misaligned samples, this often cause drift or error detection. Moreover, classifiers which used in many existing algorithms may not be good enough to discriminate the target from background such as the OAB tracker [6] [12]. In the OAB tracker, there are N number of selectors which are composed by a series of weak classifiers. When new data arrives, each of the weak classifier is updated. It can be seen that this method needs to establish a number of weak classifiers and the selection of weak classifiers directly influences the tracking results.

Trying to solve these problems, this paper proposes a novel target tracking method for online learning based on ELM. This method combines the ELM model, the target prediction method, the search mechanism and the update mechanism into a framework. Experiments have shown that the classifier used in this algorithm has a good ability to distinguish between background and the target. Furthermore, the update mechanism used in this paper analyzes sub-optimal positive samples and false positive samples such that the overall tracking performance is improved in the occluding and changing scenes.

The subsequent arrangement of this paper is as follows: Section II demonstrates the framework of this proposed algorithm. Section III shows the extraction of HOG features. ELM is explained in section IV. Section V gives the details of updating process. Section VI analyses the Experimental results. Finally, a conclusion is presented in Section VII.

II. System Architecture

The proposed method consists of three modules as shown in Fig. 1: training, tracking and classifier update.

The training module aims to train ELM by using the training set. The training set which consists of positive samples and negative samples is captured from the first frame of each sequence. Positive samples are the marked target regions and negative samples are selected randomly from background. It

This work is supported by National Natural Science Foundation of China (NSFC) under grant 61473089.

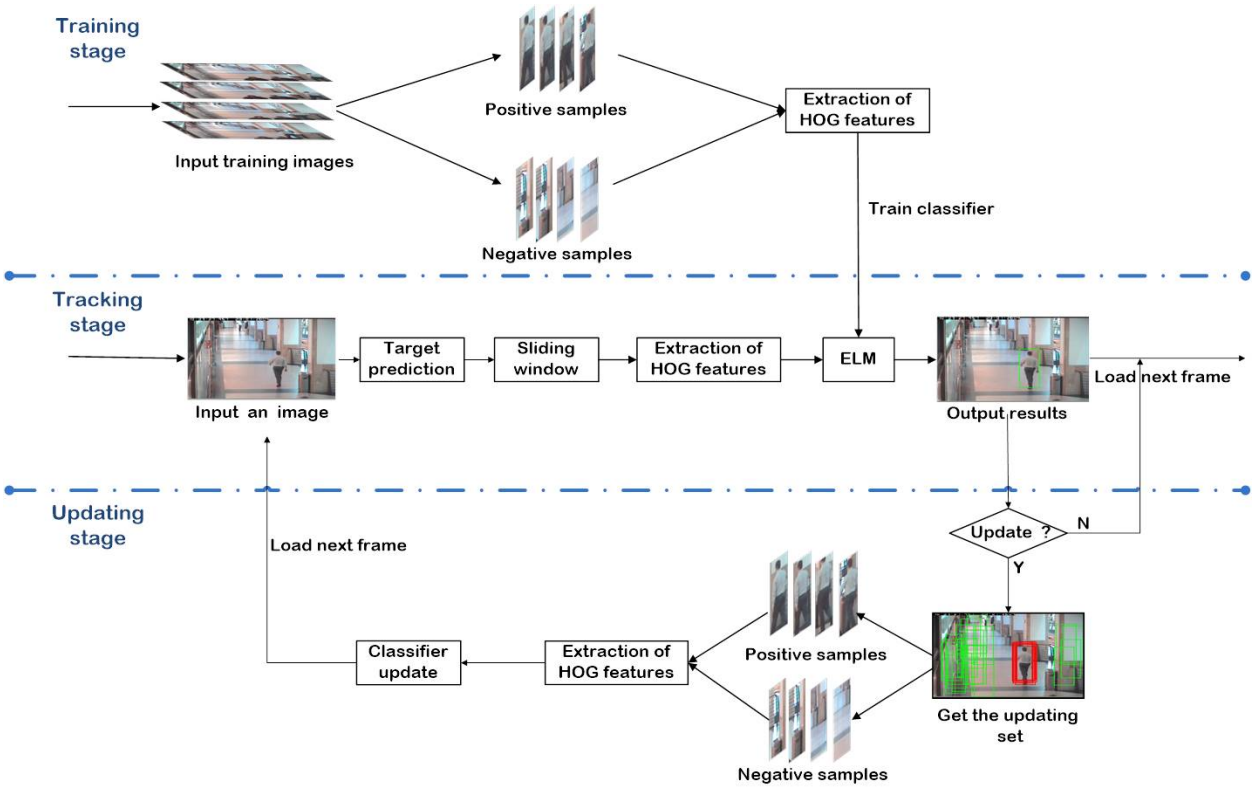

Fig. 1. Architecture of the proposed target tracking method.

is the fundamental module of target tracking. Fig. 2 presents the procession of training stage.

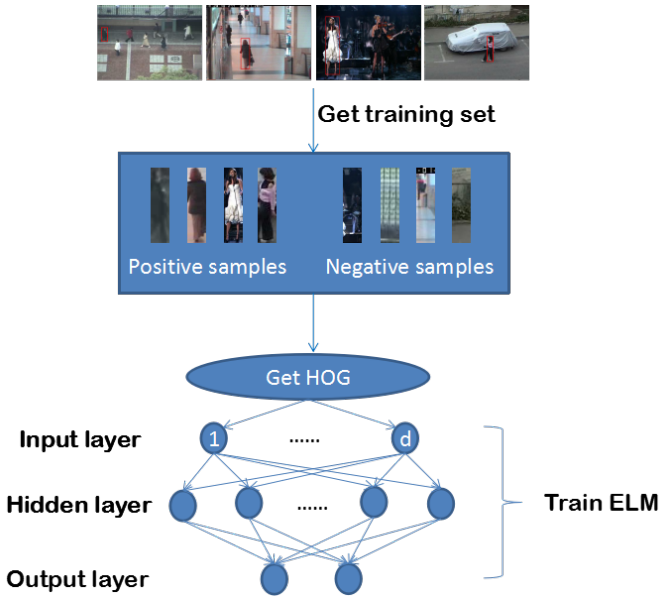

Fig. 2. The process of training stage.

The tracking stage predicts target positions and detects the targets in candidate regions. There are four steps in the tracking stage.

(1) Estimating the target position of the current frame according to its locations in previous frames. A simple motion equation used for the target prediction is determined by:

$$r_t = r_{t-\Delta t} + v * \Delta t, \tag{1}$$

where r_t is the predicted target location of the current frame, $r_{t-\Delta t}$ is the target location in the last frame and v is the speed calculated by target locations in previous frames.

(2) Using the sliding window approach to extract HOG features in candidate regions. The candidate regions are derived based on the predicted location.

(3) Putting HOG features into the trained classifier.

(4) The ELM classifier judges the window is background or a target.

In the classifier update stage, the system judges whether to update based on outputs of the current frame. If there is a need to update, the system obtains updating set and puts it into the ELM for updating. The updating set is obtained by detecting results in the current frame.

III. Extraction of HOG features

HOG [16] that first used in pedestrian detection was proposed by Dalal et al. It is mainly used to calculate the oriented gradients of the local area and describe the contour feature of the target. The extraction process of HOG features can be summarized as follows: Firstly, the image is divided into non-overlapping regions, which are called cells. The adjacent

cells form overlapping regions called blocks and the adjacent blocks form overlapping regions called windows. Secondly, the gradient orientation and amplitude of each pixel are calculated. Then a histogram for each cell is accumulated. Then normalizing the histograms of all cells in a block. Finally, the HOG feature is formed by concatenating all histograms. Fig. 3 illustrates the extraction process of HOG features.

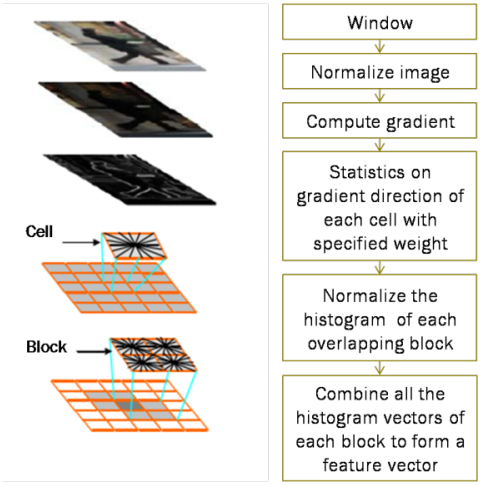

Fig. 3. Extraction of HOG features.

Compared to other feature description methods, the HOG feature has a lot of advantages. First of all, HOG is robust to the geometric and optical deformation because it only represents the local information of an image. Secondly, some small changes in the target (slight rotation, etc.) won't affect extraction results. Currently, the combination of HOG and SVM has been widely used in image recognition, especially in pedestrian detection.

IV. Extreme Learning Machine

ELM is a generalized SLFNs. It was first proposed by Huang et al. [17] [18]. ELM algorithm is better than traditional neural network algorithms because it can randomly generate the weights between the hidden layer and input layer. Therefore the training time is significantly shortened. As ELM has a good performance on the binary classification problem, this paper uses ELM as the classification model of target tracking. In this paper, the ELM structure is shown in Fig. 4

In the training phase, given N_0 training samples (x_j, t_j), where $j = 1, 2, ..., N_0$. x_j is the feature vector of the jth sample, $x_j = [x_{j1}, x_{j2}, ..., x_{jn}]^T \in R^n$. t_j is the label of the jth sample, $t_j = [t_{j1}, t_{j2}, ..., t_{jm}]^T \in R^m$. The equation for calculating the output value of the jth sample is given by

$$\sum_{i=1}^{L} \beta_i g_i(x_j) = \sum_{i=1}^{L} \beta g(w_i \cdot x_j + b_i) = o_j, j = 1, ..., N_0. \quad (2)$$

In equation (2), $g(x)$ is the activation function. This method uses the sigmoid function as the activation function, which $S(x) = \frac{1}{1+e^{-x}}$. β_i is the weight vector between the ith hidden node and the output nodes and $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$. w_i is the weight vector between the input nodes and the ith hidden node, where $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$. b_i is the bias of the ith hidden node. Equation (2) can be written as

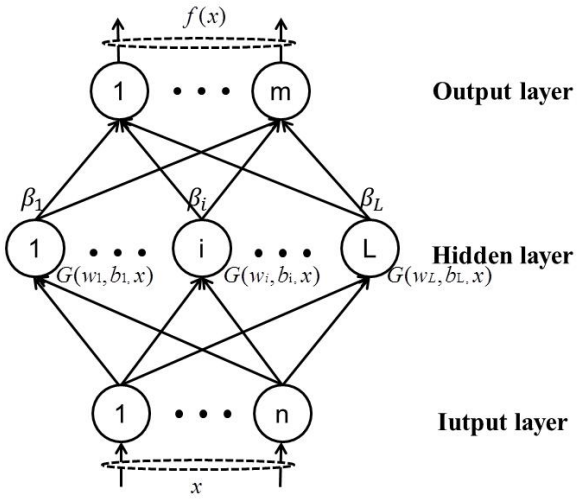

Fig. 4. The structure of ELM based model of target tracking: n is the number of input nodes. L is the number of hidden nodes. m is the number of output nodes.

$$H_0 \beta_0 = T_0, \quad (3)$$

where H_0 is the output matrix of hidden layer:

$$H_0 = \begin{bmatrix} g(w_1.x_1 + b_1) & ... & g(w_L.x_1 + b_L) \\ ... & ... & ... \\ g(w_1.x_{N_0} + b_1) & ... & g(w_L.x_{N_0} + b_L) \end{bmatrix}_{N_0 \times L}. \quad (4)$$

The ELM algorithm aims to minimize the error of outputs.

$$Minimize : \| H_0 \beta_0 - T_0 \|^2 . \quad (5)$$

Since w_i and b_i are randomly generated in the training phase, the H_0 matrix can be calculated in advance. By using the least square method, the calculation of β_0 can be written as:

$$\beta_0 = {H_0}^{\dagger} T_0, \quad (6)$$

where ${H_0}^{\dagger}$ is the pseudo-inverse matrix of H_0. Under the condition of $rank(H_0) = L$, ${H_0}^{\dagger}$ can be calculated by ${H_0}^{\dagger} = ({H_0}^T H_0)^{-1} {H_0}^T$. The equation of β_0 is as follows:

$$\beta_0 = {K_0}^{-1} {H_0}^T T_0, \quad (7)$$

where $K_0 = {H_0}^T H_0$.

V. Updating of the Classifier

The target and background may change in the tracking process, this paper therefore proposes a constructive learning algorithm based on ELM [17] [18]. The update process can

be summarized as follows: First of all, the system judges whether to update according to outputs of the classifier. If necessary, regions around the target will be selected as positive samples and regions in background will be selected randomly as negative samples. Finally, the system extracts features from the updating set and puts HOG features into the classifier for updating. Fig. 5 illustrates an example of the classifier update. In Fig. 5, positive samples are marked with red bounding boxes and negative samples are marked with green bounding boxes.

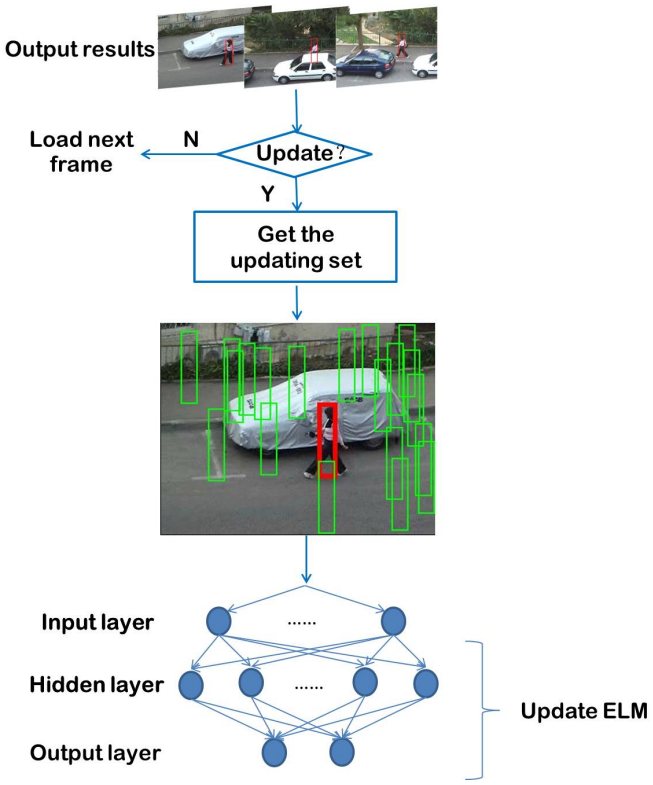

Fig. 5. An example of the classifier update.

The parameters H_0 and β_0 of the classifier are updated. The updating equations [18] of the (k+1)th tracking image are as follows:

$$P_{k+1} = P_k - P_k {H_{k+1}}^T (I + H_{k+1} P_k {H_{k+1}}^T)^{-1} H_{k+1} P_k, \quad (8)$$

$$\beta_{k+1} = \beta_k + P_{k+1} {H_{k+1}}^T ({T_{k+1}}^T - H_{k+1}\beta_k), \quad (9)$$

where $K_{k+1} = K_k + {H_{k+1}}^T H_{k+1}$ and $P_{k+1} = K_{k+1}^{-1}$.

Using the positive samples with serious occlusion or drift to update the classifier may cause error detection or target drift.. Fig. 6 shows an example of error detection in the Coke sequence. In Fig. 6, frame 38 is updated with serious occlusion. Then the occlusion is regarded as a new target in subsequent frames. The red bounding box is the detection box after updating and the blue bounding box is the ground truth. An automatic judging mechanism is also included in this method to address the samples occlusion and drift problems. The classifier won't be updated if the automatic judging results in this algorithm shows the samples are seriously occluded or shifted.

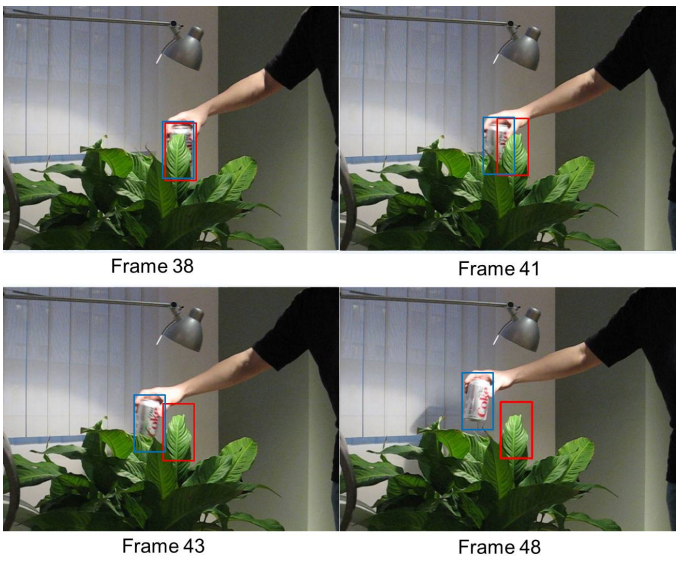

Fig. 6. An example of error detection in the Coke sequence.

VI. Experiments

A. Database

This paper uses the benchmark dataset [19] to evaluate our algorithm. All sequences used in this paper are available on the http://www.visual-tracking.net. Fig. 7 shows sequences used in our experiments and the red bounding box is the target from the first frame of each sequence.

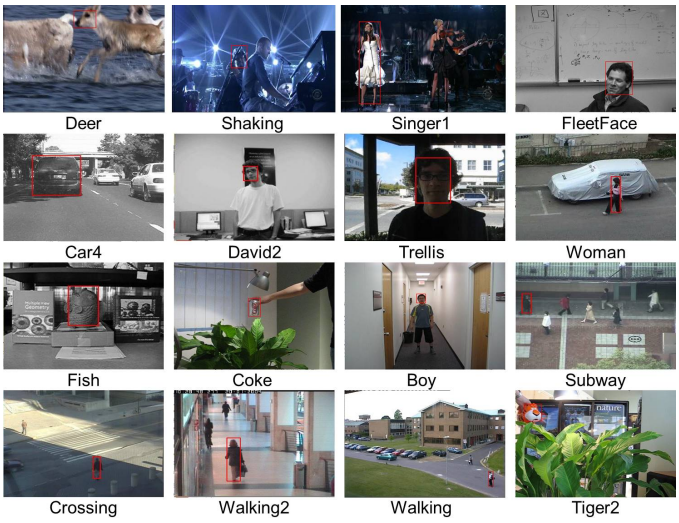

Fig. 7. Experimental sequences from the benchmark dataset.

TABLE I
TRACKING RESULTS ON THE BENCHMARK DATASET

Algorithms	success(AUC)
OURS	**59.6%**
ASLA [21]	50.9%
DFT [22]	48.7%
TLD [10]	46.0%
CSK [8]	45.5%
CXT [9]	43.4%
VTS [5]	40.6%
IVT [1]	40.0%
VTD [23]	39.4%
MIL [13]	38.9%

B. Parameters

A large number of experimental results show dimensions of HOG have little effect on the tracking results. In order to simplify the experimental parameters and improve the tracking efficiency, the dimension of HOG for each image is fixed. HOG parameters are set as follows: linear gradient voting into 9 orientation bins in 0°-360°. The window size is close to the target's size in the first frame. Moreover, it can be divisible by 4. Then block size, block stride and cell size can be calculate as follows:

$$block\ size = 0.5 * window\ size, \tag{10}$$

$$block\ stride = cell\ size = 0.5 * block\ size. \tag{11}$$

Therefore, the dimension of each HOG is 324. The number of hidden nodes is set to 300.

C. Evaluation

To quantify the performance of the experiment fairly, this paper uses success plots for our evaluation [19] [20].

Being a correct tracking, the overlap area a_0 between the tracked rectangle area B_t and the ground truth rectangle area B_{gt} must exceed the given threshold by the equation:

$$a_0 = \frac{area(B_t \cap B_{gt})}{area(B_t \cup B_{gt})}. \tag{12}$$

The success plot illustrates the percentages of correct tracking frames at the thresholds in the range of 0 to 1. Since the area under curve (AUC) of success curve is more accurate than the value at one specified threshold of plot, this paper uses AUC scores to evaluate and rank the tracking algorithms in this paper.

D. Results

The tracking performance of our proposed method and other other well-known tracking algorithms is shown in Fig. 8. Table I shows tracking results of the top 10 trackers. Obviously, the algorithm we proposed performs better than state-of-the-art methods by 8.7 percentage.

Furthermore, the benchmark dataset sequences are labelled with different properties: scale variation, out-of-plane rotation, in-of-plane rotation, fast motion, occlusion, low resolution, illumination variation, deformation, motion blur and background clutter. These attributes are the factors make the tracking extremely challenging and difficult.

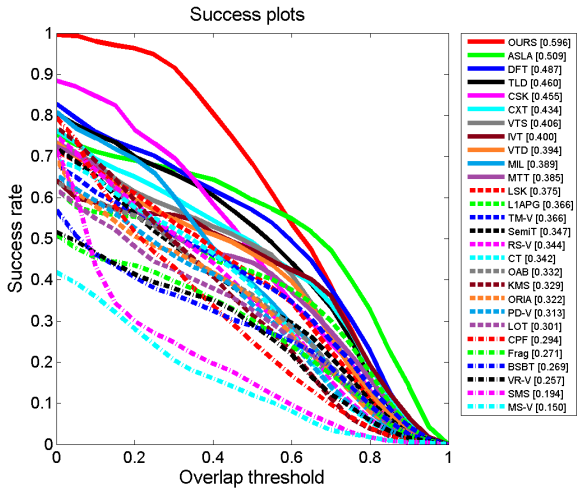

Fig. 8. Success plots of different algorithms.

Fig. 9 shows that our update mechanism does improve the performance when the target region with occlusion, illumination variation, out-of-plane rotation or in-of-plane rotation.

Due to our the use of the target prediction mechanism and the discriminative classifier, the method we proposed shows good robustness with the conditions of background clutter, motion blur and fast motion. Our algorithm has a better tracking performance in the case of background clutter, motion blur and fast motion. Fig. 10 shows success plots for sequences with attributes: fast motion, background clutter and motion blur.

In terms of deformation, low resolution and scale variation, our algorithm is comparable with other methods in Fig. 11.

VII. CONCLUSION

This paper proposes a constructive algorithm for online learning of target tracking. Different from common tracking algorithms, our tracking method does not require many classifiers but only one classifier. In addition, our tracking algorithm only retains the updating parameters. The number of positive and negative samples will not increase rapidly as time goes on. Experiments have shown that the classifier used in this algorithm has a good ability to distinguish between background and the target. Since sub-optimal positive samples and false positive samples are automatically recognized, the occlusion performance is improved. Moreover, due to the update mechanism, the method proposed in this paper can better adapt to the changes of target and background. In conclusion, the algorithm proposed in this paper outperforms other common target algorithms.

REFERENCES

[1] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

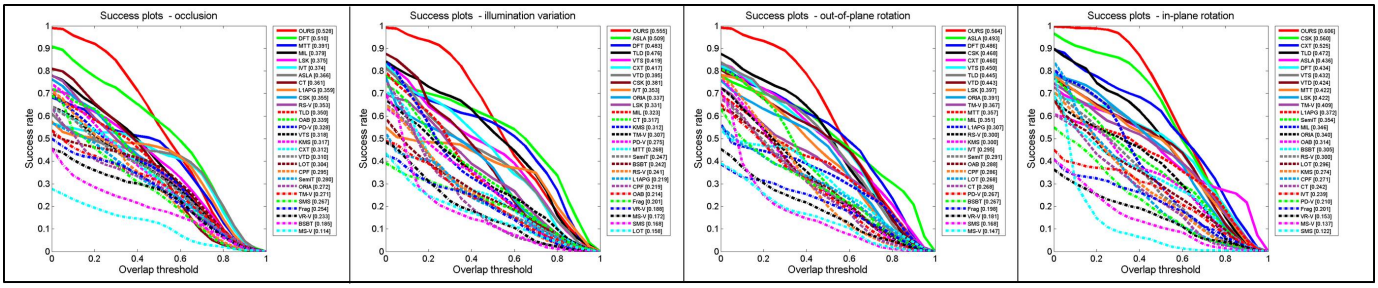

Fig. 9. Success plots for sequences with attributes: occlusion, illumination variation, out-of-plane rotation and in-of-plane rotation.

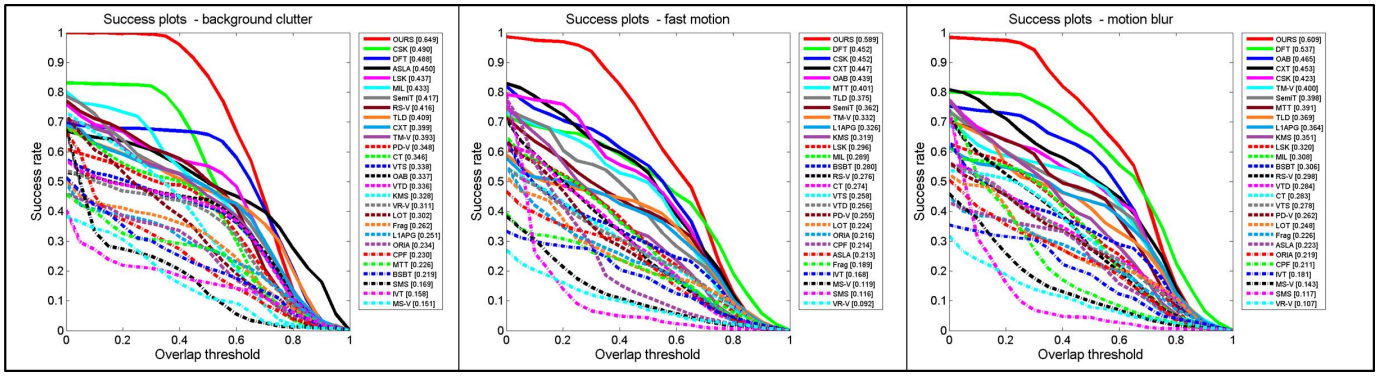

Fig. 10. Success plots for sequences with attributes: background clutter, fast motion and motion blur.

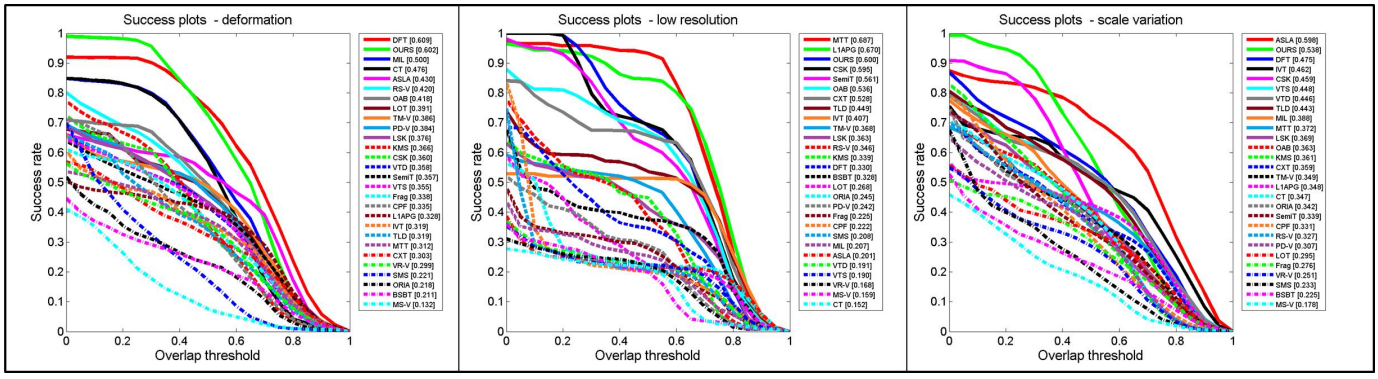

Fig. 11. Success plots for sequences with attributes: deformation, low resolution and scale variation.

[2] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1305–1312.

[3] X. Mei and H. Ling, "Robust visual tracking using l_1 minimization," in *Proceedings of IEEE International Conference on Computer Vision*, Sept 2009, pp. 1436–1443.

[4] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1830–1837.

[5] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proceedings of IEEE International Conference on Computer Vision*, Nov 2011, pp. 1195–1202.

[6] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting." in *Proceedings of the British Machine Vision Conference*, vol. 1, no. 5, 2006, p. 6.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*. Springer, 2012, pp. 702–715.

[9] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1177–1184.

[10] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 49–56.

[11] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2006, pp. 260–267.

[12] H. Grabner, J. Ŝochman, H. Bischof, and J. Matas, "Training sequential on-line boosting classifier for visual tracking," in *Proceedings of International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[13] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with

online multiple instance learning," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 983–990.

[14] S. Avidan, "Support vector tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004.

[15] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[17] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[18] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.

[19] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 2411–2418.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[21] X. Jia, H. Lu, and M. H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1822–1829.

[22] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1910–1917.

[23] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1269–1276.